

カリキュラムモデル J07-SE の概要

2009年1月
情報処理学会
ソフトウェアエンジニアリング教育委員会

1. はじめに

J07-SE は、情報専門学科カリキュラム標準 J07 におけるソフトウェアエンジニアリング領域のカリキュラムモデルである。これは情報処理学会ソフトウェアエンジニアリング教育委員会が 2006 年 4 月から 2008 年 3 月まで 2 年間に渡り議論を行って策定したものである。

本概要では、ソフトウェアエンジニアリングを取り巻く状況、J07-SE カリキュラムモデルの全体像、知識項目および科目、年次進行、今後の取り組みを概説する。

2. ソフトウェアエンジニアリングを取り巻く状況

近年、企業情報システム（エンタープライズ系システム）やソフトウェア組込み製品（組込み系システム）の発展に伴い、ソフトウェアは社会に欠くべからず存在となった。我々の生活も企業の活動も、ソフトウェア無しでは全く立ちゆかなくなってきた。

しかしその一方で、ソフトウェアに起因するシステム障害や製品リコールが多発している。特に最近では、証券取引所・銀行間取引システムのような重要な社会インフラや、人命に関わる製品にも及んでいる。このままでは、我が国の産業競争力にも消費者安全にも大きな負の影響を及ぼしてしまう。

原因については様々な議論があるが、質の高い人材を生み出すシステムが機能していないことが大きな一因である点は論を待たない。我が国のソフトウェアの多くは、きちんとソフトウェアエンジニアリング（SE）を学んでおらず、また企業でも十分な体系的教育を受けていない技術者によって作り続けられているのだ。これは極めて憂慮すべき問題である。

ソフトウェアエンジニアリングとは何か。「ソフトウェアを開発する際に駆使すべき技術」（Pressman, R.S.: Software Engineering - A Practitioner's Approach, McGraw-Hill, 2005）であり、「ソフトウェアに対する工学的原則の適用」（ISO/IEC/JTC1/SC7: ISO/IEC TR 19759:2005, Software Engineering - Guide to the Software Engineering Body of Knowledge (SWEBOK), ANSI, 2007）である。SE はコンピュータサイエンス(CS)に基づ

いているが、重点を置くところが異なる。科学者は自然法則の知識を拡張するのに対して、技術者は数多くの制約条件のもとに、役に立つ成果物を構築するために自然法則を適用しなくてはならないからだ (SWEBOK より)。しかし我が国では、CS すら学んだことのない技術者と、大学等で CS しか学ばず SE は学んでいない技術者によって多くのソフトウェアが作り続けられているという問題を抱えている。

この問題を解決するために必要なのは、多くの技術者を生み出す大学などの高等教育機関における SE 教育の質の向上である。もちろんこれまでも、意識の高い研究者は自らの講義のために実践的・先端的な SE 教育を整備してきているし、一部の大学では産学連携の SE 教育を行ってきた。いくつかの NPO 法人では SE 教育のカリキュラムや内容に対する議論が交わされている。経済産業省や文部科学省、日本経済団体連合会 (経団連) はいくつかの大学や大学院に対して助成を行っている。また経済産業省・情報処理振興機構による IT スキル標準 (ITSS) や組込みスキル標準 (ETSS)、情報システムユーザースキル標準 (UISS) などでも大学教育との関連性が議論されている。

情報処理学会では、1999 年からア krediyteeshon 委員会 SE 分科会として、SWEBOK や CCSE を始めとする SE の知識体系の調査、日本の大学学部などで実行可能なカリキュラムモデル Jpn1 の策定、(特に米国における)SE ア krediyteeshon 動向の調査などを行ってきた。これらの活動成果は平成 13 年 3 月に開催した情報処理学会「ア krediyteeshon」シンポジウムにおいて発表し、また同じく平成 13 年 3 月に発行された「ソフトウェア工学におけるア krediyteeshon に関する調査研究」調査報告書としてまとめられている。

こうした流れを踏まえ、様々な SE 教育の取り組みによって質の高い人財を生み出す仕組みを活性化できるよう、改めて情報処理学会としての SE の知識項目の整備やカリキュラムモデルの策定が強く望まれている。J07 として SE の知識項目やカリキュラムの参照モデルを提示することで、様々な SE 教育の取り組みを整理できるからである。それによって強みや弱みを明らかにでき、さらなる改善を促すことができる。もちろん、SE の学科やコースの設置が容易になることは言うまでもない。

そこで我々 SE 教育委員会は、J07 の一領域としてカリキュラムモデルを策定することとした。その内容は情報処理学会全国大会にて報告を行うとともに、J07/SE 領域の Web サイト (<http://blues.se.uec.ac.jp/j07/>) に公開した。本報告書では、以上の資料をまとめたものを報告する。

3. J07-SE の全体像

3.1 ミニマムセットのカリキュラムモデル

J07-SEは、大学などの高等教育機関の情報専門学科におけるSE教育のためのカリキュラムモデルである。まず、カリキュラムそのものではない点に注意が必要である。本カリキュラムモデルは、カリキュラムを開発するために最低限必要な知識項目、それらを教えるコア科目、演習やインターンシップの仕様、学科やコースの特色を出すためのSAS科目（System and Application Specialties: システム応用・特化型科目）科目の例、年次進行の例から構成される。学科やコースにおいてソフトウェアエンジニアリングのカリキュラムを開発するためには、本カリキュラムモデルを基にして、知識項目の把握、コア科目の理解、演習やインターンシップの実装、SAS科目の選択や開発、年次進行の策定などを行うことが必要となる。

次に、J07-SEはSEの学科やコースにおけるミニマムセットである点に注意が必要である。本カリキュラムモデルは、日本の大学が置かれている競争的環境を踏まえ、学科やコースの特色を出す余裕を持たせるために、最低限必要な知識項目に絞ってコア科目を列挙している。そのため、学科やコースの特色を反映した科目を追加しなくてはならない。例として、エンタープライズ系システム向けSEカリキュラムや組込みシステム向けSEカリキュラム、Webシステム向けSEカリキュラムなどの適用ドメイン的な特色を反映した学科やコース、要求開発やアーキテクチャ設計、実装、検証と妥当性確認、マネジメントなど技術的な特色を反映した学科やコースなどが考えられる。J07-SEでは、カリキュラム開発者のためにSAS科目として、学科やコースの特色を反映させるための追加科目を例示している。また特色を出しやすいように、演習などについては内容の詳細に踏み込まず、仕様を提示するに留めている。

また真に実践的で充実した教育を行うためには、学部教育に許された時間数では少ないと言わざるを得ない。大学院との連携教育が可能な学科やコースは、ミニマムセットであるJ07-SEを拡張することで大学院修士課程も含めた6年間のカリキュラムモデルを構築するとよい。その際にはSE教育委員会の以前の成果物であるJpn1カリキュラムモデルが参考になるだろう。

3.2 コンセプト

SEは、その名の通り「エンジニアリング」である。エンジニアリング教育には、主に2つの特性が求められる。一つは“実践的”であり、もう一つは“骨太”である。

従来から大学で行われている情報系の教育は、その出自や理念などにより、理論的なものが多いと言われている。しかし産業界からは、開発現場であまり役に立たないという批判も多い。そこでJ07-SEは、産業界のニーズに応えるために従来の基礎一辺倒のカリキュラムから、実践的内容にも踏み込んだカリキュラムモデルを指向することとした。まず、プログラミング言語の習得に留まらず、開発の全体像である要求開発から検証と妥当性確

認、保守、プロセスやマネジメントといった開発ライフサイクルを網羅するカリキュラムモデルとなっている。次に、理論の理解に留まらず、開発の目的である品質・生産性・コストといった要因を重視するカリキュラムモデルとなっている。また、個人に閉じた作業に留まらず、開発に必要なマネジメントやコミュニケーション、チームダイナミクスなどを学び実践するカリキュラムモデルとなっている。さらに、開発者のモチベーションや、不具合につながる開発者のヒューマンエラーといった心理的側面も取り扱っている。

一方で J07-SE は、SE の技術進化の速さを鑑みるに近視眼的な技術訓練では不十分であると考え、骨太なカリキュラムモデルを指向した。特に開発現場寄りの産業界からは、すぐに現場で使える技術を求められることもある。しかし、SE において重要なのは技術の使い方よりも“ものの考え方”そのものである。そこで、モデル化を習得することで「捉える力」や「考える力」、「表現する力」などを、検証と妥当性確認やプロセス改善を習得することで「問題発見能力」や「問題解決能力」などを、プロセスやマネジメントを習得することで「段取り力」や「調整力」などを涵養することを指向した。またソフトウェアに留まらない一般的な工学原則も学ぶこととした。これによって、技術を単に理解するだけでなく、臨機応変に応用でき中長期的に付加価値を生み続けられる技術者の育成が期待できる。

J07-SE の特徴は、こうした内容を PBL (Project Based Learning) のみに担わせるべきではないと意図している点にある。PBL はその性格上、基本概念や技術を習得する際に、PBL で採用した開発やマネジメントのスタイルに基づく一面的な理解になりかねない。そこで J07-SE では、講義においてさまざまな基本概念や技術を提示することで、PBL が採用するスタイル以外のさまざまな開発やマネジメントのスタイルの特徴を多面的に深く理解することを指向した。

また、卒業研究によって SE の技術やマネジメントの本質に触れ、技術者として巣立った後に SE の技術やマネジメントを改善し発展させ新たに創造することも期待している。したがって J07-SE は、実践的な教育を実施するという意味で PBL を卒業研究に代えるという構成の学科やコースも許容する一方で、PBL に加えて卒業研究を行う学科やコースも許容している。J07-SE を参照する学科やコースは、ぜひ講義での理解と PBL での実践を踏まえた質の高い卒業研究指導を目指して欲しい。

3.3 カリキュラムモデルの構造

J07-SE は、講義科目と実習科目、SAS 科目、インターンシップ、卒業論文から構成される。講義科目はさらに、情報科学基礎科目と SE 科目に分けられる。情報科学基礎科目とは、論理と計算理論、離散数学、オペレーティングシステム基礎・データベース基礎といった SE の基礎となる科目である。SE 科目とは、ソフトウェア構築やソフトウェア設計、検証

と妥当性確認、開発マネジメントといった SE 技術を扱う科目である。

実習科目は、プログラミング入門、プログラミング基礎実習、プログラミング応用実習、ソフトウェア開発実習から構成される。プログラミング入門では、プログラミング言語や開発環境の習得、データ構造とアルゴリズムの理解と実装を行う。教養系で実施する一般的なプログラミング入門を想定している。プログラミング基礎実習では、概略設計内容に対する個人でのプログラム開発と単体・統合テストを行う。プログラミング応用実習では、要求仕様に対する個人ないしグループでのアーキテクチャ設計、ソフトウェア設計、プログラム開発、単体・統合・システムテストを行う。ソフトウェア開発実習では、システムへの要望に対するグループでのシステム開発とプロジェクトマネジメントを行う。エンタープライズ系や組込み系など学科やコースの特色に合わせて、かなり具体的なシステムを想定した実習となる。

SAS (System and Application Specialties) 科目は、学科やコースの特色を反映させるための追加科目である。J07-SE では、システムのドメイン、システムの特長、システムの構造、開発技術による SAS 科目を例示している。ただし科目名の提示に留め、詳細は学科やコースで定める必要がある。

インターンシップは、民間企業などとの連携により、現実のソフトウェア開発プロジェクトを教材とした実践的な学習を行う科目である。単なる例題にとどまらない現実のソフトウェア開発作業に関わることにより、納期、予算、組織といった実社会の制約を踏まえたソフトウェア開発の実際について学ぶ。目的意識を持った内容の濃いインターンシップとするために、事前報告書を作成することでインターンシップの目的を深耕し、事後報告書や発表によって内容を咀嚼するだけでなく、インターンシップ受け入れ先からも評価を受ける必要がある。

卒業研究は、在学中に修得したソフトウェア工学に関する知識や技術のすべてを、初めての新しい課題に集中して独自の解を求め、その研究成果を論文または制作物の形にまとめ、発表する科目である。これを通して、研究の推進方法、関連研究の調査による知識の進歩、および、新しい独創的な考えを展開してゆく能力を養成する。

4. J07-SE の知識項目および科目

4.1 知識項目

カリキュラムモデルにまず必要となるのは、教えるべき知識項目の整理である。SE 分野には SWEBOOK (Software Engineering Body of Knowledge) などいくつかの知識項目が定義されているが、J07-SE では J07 の方針を踏襲し、CCSE2004 (Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering / A Volume of the

Computing Curricula Series, August 23, 2004) を参照し、科目として取り扱いやすい単位として再体系化を行っている。表 1 に知識項目の第 2 カテゴリまでを示す。付録 1 に J07-SE の知識項目を示す。知識項目の ID は CCSE と一致させ、CCSE2004 に存在しない知識項目は、N/A で表している。

表 1 J07-SE の知識項目 (第 2 カテゴリまで)

コンピュータとソフトウェアの基礎		
CMP		コンピュータ基礎
	CMP.cf	コンピュータ科学基礎
N/A	N/A	ソフトウェア工学の基礎
確率・統計		
FND		数理基礎・工学基礎
	FND.mf	数理基礎
	FND.ef	ソフトウェアのための工学基礎
離散数学		
FND		数理基礎・工学基礎
	FND.mf	数理基礎
プログラミング基礎		
CMP		コンピュータ基礎
	CMP.cf	コンピュータ科学基礎
	CMP.ct	構築技術
	CMP.tl	構築のためのツール
論理と計算理論		
FND		数理基礎・工学基礎
	FND.mf	数理基礎
オペレーティングシステム基礎・データベース基礎		
CMP		コンピュータ基礎
	CMP.cf	コンピュータ科学基礎
ネットワーク基礎		
CMP		コンピュータ基礎
	CMP.cf	コンピュータ科学基礎
工学基礎		
CMP		コンピュータ基礎
	CMP.cf	コンピュータ科学基礎
FND		数理基礎・工学基礎

	FND.ef	ソフトウェアのための工学基礎
PRF		プロフェッショナルプラクティス
	PRF.pr	プロフェッショナリズム
MAA		ソフトウェアのモデリングと分析
	MAA.af	モデルの分析の基礎
ソフトウェア構築		
CMP		コンピュータ基礎
	CMP.cf	コンピュータ科学基礎
	CMP.ct	構築技術
	CMP.tl	構築のためのツール
DES		ソフトウェア設計
	DES.con	設計に用いられる概念
	DES.dd	詳細設計
モデル化と要求開発		
MAA		ソフトウェアのモデリングと分析
	MAA.md	モデリングの基礎
	MAA.tm	モデルの種類
	MAA.rv	要求の評価
ソフトウェアアーキテクチャ		
DES		ソフトウェア設計
	DES.con	設計に用いられる概念
	DES.str	設計のパラダイム
	DES.ar	アーキテクチャ設計
	DES.ste	設計の支援ツールと評価
ソフトウェア設計		
DES		ソフトウェア設計
	DES.dd	詳細設計
検証と妥当性確認		
VAV		検証と妥当性確認(V&V)
	VAV.fnd	V&Vの用語と基礎
	VAV.rev	レビュー
	VAV.tst	テスト
	VAV.par	不具合の分析と報告
形式手法		
CMP		コンピュータ基礎

	CMP.fm	形式手法
MAA		ソフトウェアのモデリングと分析
	MAA.md	モデリングの基礎
ソフトウェアプロセスと品質		
FND		数理基礎・工学基礎
	FND.ec	ソフトウェアのためのエンジニアリングエコノミクス
PRF		プロフェッショナルプラクティス
	PRF.pr	プロフェッショナリズム
EVO		ソフトウェアの進化や保守
	EVO.pro	進化や保守のプロセス
	EVO.ac	進化や保守のアクティビティ
PRO		ソフトウェア開発プロセス
	PRO.con	プロセスの基礎
	PRO.imp	プロセスの実装
QUA		ソフトウェア品質
	QUA.cc	ソフトウェア品質の概念と文化
	QUA.std	ソフトウェア品質に関する標準
	QUA.pro	ソフトウェア開発プロセスの改善
	QUA.pca	プロセスの保証
	QUA.pda	製品の保証
ヒューマンファクター		
CMP		コンピュータ基礎
	CMP.cf	コンピュータ科学基礎
DES		ソフトウェア設計
	DES.hci	ヒューマン・コンピュータ・インタフェース(HCI)設計
VAV		検証と妥当性確認(V&V)
	VAV.hct	HCIのテストと評価
開発マネジメント		
FND		数理基礎・工学基礎
	FND.ec	ソフトウェアのためのエンジニアリングエコノミクス
PRF		プロフェッショナルプラクティス
	PRF.psy	グループダイナミクス / 心理学
	PRF.com	(SEに特化した)コミュニケーションスキル
	PRF.pr	プロフェッショナリズム
MAA		ソフトウェアのモデリングと分析

	MAA.rfd	要求分析の基礎
	MAA.er	要求の獲得
	MAA.rsd	要求の仕様化と文書化
MGT		ソフトウェア開発のマネジメント
	MGT.con	マネジメントの基礎
	MGT.pp	プロジェクトの計画
	MGT.per	プロジェクトのメンバと組織
	MGT.ctl	プロジェクトのコントロール
	MGT.cm	ソフトウェア構成管理

4.2 情報科学基礎科目

J07-SE では前述した知識項目を基に、8つの情報科学基礎科目を提示した。表2に情報科学基礎科目の一覧を示す。また付録2に情報科学基礎科目の内容を示す。なお確率統計については教養系で実施する講義を想定しているが、各種SE技術の背景・基礎という観点で習得の必要な内容をカリキュラムモデルに含めて策定した。プログラミング基礎については、学科やコースの特色に応じたプログラミング言語を対象として科目を開発すべきである。本カリキュラムモデルでは、例としてC言語を用いた科目を提示している。

表2 J07-SE の情報科学基礎科目の一覧

コンピュータとソフトウェアの基礎
確率・統計
離散数学
プログラミング基礎
論理と計算理論
オペレーティングシステム基礎・データベース基礎
ネットワーク基礎
工学基礎

4.3 SE 科目

J07-SE では前述した知識項目を基に、9つのSE科目を提示した。表3にSE科目の一覧を示す。また付録3にSE科目の内容を示す。

表3 J07-SEのSE科目の一覧

ソフトウェア構築
モデル化と要求開発
ソフトウェアアーキテクチャ
ソフトウェア設計
検証と妥当性確認
形式手法
ソフトウェアプロセスと品質
ヒューマンファクター
開発マネジメント

4.4 実習科目とインターンシップ、卒業研究

J07-SEでは、4つの実習科目とインターンシップ、卒業研究を提示した。表4に実習科目、インターンシップ、卒業研究の一覧を示す。また付録4に実習科目、インターンシップ、卒業研究の内容を示す。なおプログラミング入門については、教養系で実施する一般的な演習を想定しているため、科目内容は策定していない。またソフトウェア開発実習については、例としてエンタープライズ系と組込み系の2種を示す。

表4 J07-SEの実習科目、インターンシップ、卒業研究の一覧

プログラミング入門
プログラミング基礎実習
プログラミング応用実習
ソフトウェア開発実習
インターンシップ
卒業研究

4.5 SAS科目

J07-SEではSAS科目の例を提示した。SAS科目の例は、システムのドメイン、システム特性、システムの構造、開発技術などに分類している。ただしJ07-SEではSAS科目例の内容は定めていない。表5にSAS科目の例一覧を示す。

表5 J07-SEのSAS科目の例の一覧

システムのドメインによるSAS科目
Webシステム(ネットワーク中心・Webシステム)
企業情報システム・ERP(情報システムとデータ処理)
金融・電子商取引システム
物流・小売システム
通信・ネットワークシステム
マルチメディア・エンターテインメントシステム(ゲーム・娯楽システム)
ユビキタスシステム(モバイルシステム)
航空・車システム
工業プロセス制御システム
生命医学システム(バイオメディカルシステム)
科学技術計算システム
システムの特性によるSAS科目
高信頼性・高可用性システム/組込み(ディペンダブルシステム)
高信頼性・高可用性システム/エンタープライズ(ディペンダブルシステム)
高セキュリティ情報システム
高安全性組込みシステム
システムの構造によるSAS科目
ハードウェア制御・リアルタイムシステム(組込み・リアルタイムシステム)
トランザクションシステム
制御モデル開発
経営情報システム
レガシーシステムと派生開発
OSSによるシステム開発
エージェント・人工知能システム
開発技術によるSAS科目
ソフトウェア構築・応用
モデル化と要求開発・応用
ソフトウェアアーキテクチャ・応用
ソフトウェア設計・応用
検証と妥当性確認・応用
形式手法・応用
ソフトウェアプロセスと品質・応用

ヒューマンファクター・応用
開発マネジメント・応用

5. 年次進行の例

J07-SE ではカリキュラムモデルとして、年次進行の例を提示した。この年次進行は、3つの視点からボトムアップ的な年次進行を意図している。一つ目として、前半の年次で情報科学基礎科目を学ぶことでシステムや計算機、情報技術を理解し、後半の年次にSE科目を学ぶ素地を養う。二つ目として、学生にとって具体的にイメージしやすいプログラムコードの作成から講義を始め、モデル化と要求開発や形式手法など抽象度の高い講義を後半に行うことで、抽象化の必要性を理解させ適切な抽象化を習得させる。三つ目として、構築など個人での作業を前半に習得させ、プロセスやマネジメントなどチームやプロジェクトでの作業を後半に学ぶことで、チームやプロジェクトでの開発に必要な基本概念や技法の重要性を認識させる。これらによって、スムーズかつ適切な理解を促すことが期待できる。

ただしJ07-SEの年次進行例はあくまで例示であり、カリキュラム開発時には学科やコースの特色に応じて適切な年次進行を策定する必要がある。表6にJ07-SEの年次進行の例を示す。なお例において、講義科目と実習科目の履修に3年半で約500時間かけることを想定している。

表6 J07-SEの年次進行の例

1年前期
コンピュータとソフトウェアの基礎
1年後期
確率・統計
離散数学
プログラミング基礎
プログラミング入門
2年前期
論理と計算理論
オペレーティングシステム基礎・データベース基礎
ソフトウェア構築
2年後期
ネットワーク基礎

モデル化と要求開発
ソフトウェア設計
プログラミング基礎実習
3年前期
ソフトウェアアーキテクチャ
検証と妥当性確認
ソフトウェアプロセスと品質
プログラミング応用実習
3年夏期休暇
インターンシップ
3年後期
形式手法
開発マネジメント
ヒューマンファクター
ソフトウェア開発実習
4年前期
工学基礎
卒業研究
4年前期
卒業研究

6. おわりに

本概要では、ソフトウェアエンジニアリングを取り巻く状況、J07-SEカリキュラムモデルの全体像、知識項目および科目、年次進行を概説した。今後の取り組みとしては、J07-SEを基にしたカリキュラム例の策定、複数の大学や大学院のSEカリキュラムの比較、ITSS/ETSS/UISSなどとの連携が考えられる。多面的な開発やマネジメントのスタイルを理解できるようなPBLの例を提示することも必要だろう。また産業界や学会からの意見を募り、さらに実践的かつ骨太なカリキュラムを目指して改善していくとともに、大学院修士課程を含む6年でのカリキュラムモデルの検討も視野に入れたい。

J07-SEを策定した情報処理学会ソフトウェアエンジニアリング教育委員会のメンバを以下に示す。このメンバは産学間のバランスを保ち実践と理論の協調を図るとともに、世代間のバランスを保ち継続的な活動を可能にすることを意図している。また情報処理学会ソフトウェア工学研究会（SIGSE）と密接に連携しており、先端的な技術動向を広く普及啓

蒙ることが視野に入っている。

委員長：阿草清滋（名古屋大学）

幹事：羽生田栄一（豆蔵、情報処理学会ソフトウェア工学研究会主査）、玉井哲雄（東大）、佐伯元司（東工大）、深澤良彰（早大）、榊原彰（日本 IBM）、沢田篤史（南山大）、鷺崎弘宜（早大）、西康晴（電通大）

委員：青木利晃（北陸先端大）、飯田元（奈良先端大）、石川冬樹（NII）、位野木万里（東芝ソリューション）、大西淳（立命館大）、大森久美子（NTT 情報流通基盤総合研究所）、片山徹郎（宮崎大）、小林隆志（名古屋大）、田口研治（NII）、野中誠（東洋大）、藤井拓（オージス総研）、松下誠（阪大）、山本里枝子（富士通研）

付録1 J07-SE の知識項目

コンピュータとソフトウェアの基礎			
CMP			コンピュータ基礎
	CMP.cf		コンピュータ科学基礎
		CMP.cf.5	コンピュータの構造
		CMP.cf.6	システムの基礎
N/A	N/A	N/A	ソフトウェア工学の基礎
確率統計			
FND			数理基礎・工学基礎
	FND.mf		数理基礎
		FND.mf.6	離散確率
		FND.mf.9	数値誤差と精度
	FND.ef		ソフトウェアのための工学基礎
		FND.ef.2	統計解析(検定と推定、回帰分析、相関など)
離散数学			
FND			数理基礎・工学基礎
	FND.mf		数理基礎
		FND.mf.5	グラフとツリー
		FND.mf.10	数論
		FND.mf.11	代数構造
プログラミング基礎			
CMP			コンピュータ基礎
	CMP.cf		コンピュータ科学基礎
		CMP.cf.1	プログラミング基礎(制御とデータ、型付け、再帰)
		CMP.cf.2	アルゴリズムとデータ構造、データ表現(静的・動的)、複雑性
		CMP.cf.4	抽象化(カプセル化や階層化など)
		CMP.cf.9	プログラミング言語の基礎
		CMP.cf.13	プログラミング言語の意味論
	CMP.ct		構築技術
		CMP.ct.2	コードの再利用とライブラリ
		CMP.ct.4	パラメータ化と汎化
		CMP.ct.5	アサーション、契約による設計(DbC)、防御的プログラミング
		CMP.ct.6	エラーハンドリング、例外処理、フォールトトレラント
	CMP.tl		構築のためのツール

		CMP.tl.1	開発環境
		CMP.tl.2	GUI構築ツール
		CMP.tl.3	単体テストツール
		CMP.tl.4	アプリケーション指向言語(スクリプト言語、ビジュアル言語、ドメイン特化言語、マークアップ言語、マクロなど)
		CMP.tl.5	プロファイリング・パフォーマンス分析・スライシングのツール
論理と計算理論			
FND			数理基礎・工学基礎
	FND.mf		数理基礎
		FND.mf.2	論理学基礎(命題、述語)
		FND.mf.3	証明技法
		FND.mf.4	数え上げ基礎
		FND.mf.7	有限状態機械と正規表現
		FND.mf.8	文法
オペレーティングシステム基礎・データベース基礎			
CMP			コンピュータ基礎
	CMP.cf		コンピュータ科学基礎
		CMP.cf.10	オペレーティングシステムの基礎
		CMP.cf.11	データベースの基礎
ネットワーク基礎			
CMP			コンピュータ基礎
	CMP.cf		コンピュータ科学基礎
		CMP.cf.12	ネットワーク通信の基礎
工学基礎			
CMP			コンピュータ基礎
	CMP.cf		コンピュータ科学基礎
		CMP.cf.3	問題解決技法
		CMP.cf.6	システムの基礎
FND			数理基礎・工学基礎
	FND.ef		ソフトウェアのための工学基礎
		FND.ef.1	統計的技法と実験的技法(CPUやメモリの利用に対する測定法)
		FND.ef.3	測定とメトリクス
		FND.ef.4	システム特性(セキュリティ、安全性、パフォーマンス、スケーラビリティ、機能競合など)

		FND.ef.5	工学的設計の基本概念(問題の定式化、別解?、フィージビリティスタディなど)
		FND.ef.6	測定の理論(意味のある測定の基準など)
		FND.ef.7	ソフトウェア工学以外の分野における工学原理(材料強度、デジタル回路の原理、論理設計、熱力学の基礎など)
PRF			プロフェッショナルプラクティス
	PRF.pr		プロフェッショナリズム
		PRF.pr.2	倫理綱領とプロフェッショナルとしての行動
MAA			ソフトウェアのモデリングと分析
	MAA.af		モデルの分析の基礎
		MAA.af.1	まとまりの分析(完全性、一貫性、ロバスト性など)
		MAA.af.2	正当性の分析(静的解析、シミュレーション、モデルチェッキングなど)
		MAA.af.3	品質特性(非機能特性)の分析(安全性、セキュリティ、性能、根本原因分析など)
		MAA.af.4	優先順位付け、トレードオフ分析、要求のためのリスク分析および影響解析
		MAA.af.5	トレーサビリティ
ソフトウェア構築			
CMP			コンピュータ基礎
	CMP.cf		コンピュータ科学基礎
		CMP.cf.9	プログラミング言語の基礎
	CMP.ct		構築技術
		CMP.ct.1	APIの設計と利用
		CMP.ct.2	コードの再利用とライブラリ
		CMP.ct.3	オブジェクト指向パラダイムにおける実行時のトピック(ポリモルフィズム、ダイナミックバインディングなど)
		CMP.ct.4	パラメータ化と汎化
		CMP.ct.5	アサーション、契約による設計(DbC)、防御的プログラミング
		CMP.ct.6	エラーハンドリング、例外処理、フォールトトレラント
		CMP.ct.7	状態ベースおよびテーブル駆動の構築技法
		CMP.ct.8	実行時コンフィグレーションと国際化
		CMP.ct.9	文法ベースの入力処理(パース処理)
		CMP.ct.10	並列処理の基本要素(セマフォ、モニターなど)
		CMP.ct.11	ミドルウェア(コンポーネントとコンテナ)
		CMP.ct.12	分散ソフトウェアのための構築技術

		CMP.ct.13	組込みシステムの構築とハードウェア・ソフトウェア協調設計
		CMP.ct.14	ホットスポット分析とパフォーマンスチューニング
		CMP.ct.15	プラットフォーム標準(POSIXなど)
		CMP.ct.16	テストファーストプログラミング
	CMP.tl		構築のためのツール
		CMP.tl.1	開発環境
		CMP.tl.2	GUI構築ツール
		CMP.tl.3	単体テストツール
		CMP.tl.4	アプリケーション指向言語(スクリプト言語、ビジュアル言語、ドメイン特化言語、マークアップ言語、マクロなど)
		CMP.tl.5	プロファイリング・パフォーマンス分析・スライシングのツール
DES			ソフトウェア設計
	DES.con		設計に用いられる概念
		DES.con.1	設計という概念の定義
		DES.con.2	基本的な設計の考慮事項(データの永続性、ストレージマネジメント、例外など)
		DES.con.3	複数のソフトウェア開発ライフサイクルにおける設計の関係
		DES.con.4	設計の原則(情報隠蔽、凝集度と結合度)
		DES.con.5	設計と要求との競合
		DES.con.6	品質特性の設計(信頼性、ユーザビリティ、性能、テスト容易性、フォールトトレラント性)
		DES.con.7	設計におけるトレードオフ
		DES.con.8	アーキテクチャのスタイル、パターン、再利用
	DES.dd		詳細設計
		DES.dd.3	コンポーネント設計
		DES.dd.4	コンポーネントとシステムのインタフェース設計
モデル化と要求開発			
MAA			ソフトウェアのモデリングと分析
	MAA.md		モデリングの基礎
		MAA.md.1	モデリングの原則(分解、抽象化、汎化、投影 / ビュー、明快性、形式的アプローチの利用など)
	MAA.tm		モデルの種類
		MAA.tm.1	情報やデータのモデリング(ERD、クラス図など)
		MAA.tm.2	振る舞いのモデリング(構造化分析、状態遷移図、ユースケース分析、イ

			インタラクション図、FMEA、FTAなど)
		MAA.tm.3	構造のモデリング(アーキテクチャなど)
		MAA.tm.4	ドメインのモデリング(ドメインエンジニアリング・アプローチなど)
		MAA.tm.5	機能のモデリング(コンポーネント図など)
		MAA.tm.6	エンタープライズシステムのモデリング(ビジネスプロセス、組織、ゴールなど)
		MAA.tm.7	組込みシステムのモデリング(リアルタイムスケジュール分析、外部インタフェース分析など)
		MAA.tm.8	要求競合のための分析(機能競合、品質の家、ビューポイント分析など)
		MAA.tm.9	アナリシスパターン(プロブレムフレーム、仕様の再利用など)
	MAA.rv		要求の評価
		MAA.rv.1	レビューとインスペクション
		MAA.rv.2	要求の評価のためのプロトタイピング(累積型プロトタイピング)
		MAA.rv.3	受け入れテストの設計
		MAA.rv.4	品質特性の評価
ソフトウェアアーキテクチャ			
DES			ソフトウェア設計
	DES.con		設計に用いられる概念
		DES.con.1	設計という概念の定義
		DES.con.2	基本的な設計の考慮事項(データの永続性、ストレージマネジメント、例外など)
		DES.con.3	複数のソフトウェア開発ライフサイクルにおける設計の関係
		DES.con.4	設計の原則(情報隠蔽、凝集度と結合度)
		DES.con.5	設計と要求との競合
		DES.con.6	品質特性の設計(信頼性、ユーザビリティ、性能、テスト容易性、フォールトトレラント性)
		DES.con.7	設計におけるトレードオフ
		DES.con.8	アーキテクチャのスタイル、パターン、再利用
	DES.str		設計のパラダイム
		DES.str.1	機能指向による設計
		DES.str.2	オブジェクト指向による設計
		DES.str.3	データ構造を中心とした設計
		DES.str.4	アスペクト指向による設計
	DES.ar		アーキテクチャ設計

		DES.ar.1	アーキテクチャスタイル(パイプアンドフィルタ、レイヤード、トランザクション中心、ピアツーピア、publish/subscribe、イベント駆動、クライアントサーバなど)
		DES.ar.2	アーキテクチャで考慮すべき様々な特性間のトレードオフ
		DES.ar.3	ソフトウェアアーキテクチャで考慮すべきハードウェア
		DES.ar.4	アーキテクチャにおける要求のトレーサビリティ
		DES.ar.5	ドメイン特化アーキテクチャおよびプロダクトライン開発
		DES.ar.6	アーキテクチャのための記法(アーキテクチャ上のビューポイントと表現、コンポーネント図など)
	DES.ste		設計の支援ツールと評価
		DES.ste.1	設計支援ツール(アーキテクチャ、静的解析、動的評価など)
		DES.ste.2	設計上の特性の測定(結合度、凝集度、情報隠蔽、関心事の分離など)
		DES.ste.3	設計のメトリクス(アーキテクチャ上の要因、変換、一般的な使い方におけるメトリクス)
		DES.ste.4	フォーマルメソッドによる設計の分析
ソフトウェア設計			
DES			ソフトウェア設計
	DES.dd		詳細設計
		DES.dd.1	詳細設計の技法(SSA/SD、JSD、OODなど)
		DES.dd.2	デザインパターン
		DES.dd.3	コンポーネント設計
		DES.dd.4	コンポーネントとシステムのインタフェース設計
		DES.dd.5	設計の記法(クラス図とオブジェクト図、UML、状態遷移図など)
検証と妥当性確認			
VAV			検証と妥当性確認(V&V)
	VAV.fnd		V&Vの用語と基礎
		VAV.fnd.1	V&Vの目的と制約
		VAV.fnd.2	V&Vの計画
		VAV.fnd.3	V&Vの戦略のドキュメント化(テストなど)
		VAV.fnd.4	メトリクスと測定(信頼性、ユーザビリティ、性能など)
		VAV.fnd.5	V&Vに関連する活動
	VAV.rev		レビュー
		VAV.rev.1	机上チェック
		VAV.rev.2	ウォークスルー
		VAV.rev.3	インスペクション

	VAV.tst		テスト
		VAV.tst.1	単体テスト
		VAV.tst.2	例外のハンドリング(例外のハンドリングを発生させるテストの設計、よい例外ハンドリングの設計)
		VAV.tst.3	カバレッジ分析(命令網羅、分岐網羅、基本パス網羅、複合条件網羅、データフローパス網羅など)
		VAV.tst.4	ブラックボックスの技法
		VAV.tst.5	結合テスト
		VAV.tst.6	ユースケースや顧客シナリオによるテストケースの設計
		VAV.tst.7	操作プロファイルによるテスト
		VAV.tst.8	システムテストと受け入れテスト
		VAV.tst.9	品質特性に関連するテスト(ユーザビリティ、セキュリティ、互換性、アクセシビリティなど)
		VAV.tst.10	回帰テスト
		VAV.tst.11	テストツール
	VAV.par		不具合の分析と報告
		VAV.par.1	不具合報告書の分析
		VAV.par.2	デバッグ/不具合切り分けの技法
		VAV.par.3	欠陥分析
		VAV.par.4	不具合の追跡
形式手法			
CMP			コンピュータ基礎
	CMP.fm		形式手法
		CMP.fm.1	抽象機械の適用(SDL、Paisleyなど)
		CMP.fm.2	仕様記述言語および技法の適用(ADM、B、CPS、VDMなど)
		CMP.fm.3	仕様からのソースコードの自動生成
		CMP.fm.4	プログラム導出
		CMP.fm.5	候補となる実装の解析
		CMP.fm.6	異なる実装と仕様とのマッピング
		CMP.fm.7	詳細化
		CMP.fm.8	正当性の検証
MAA			ソフトウェアのモデリングと分析
	MAA.md		モデリングの基礎
		MAA.md.1	モデリングの原則(分解、抽象化、汎化、投影 / ビュー、明快性、形式的アプローチの利用など)

		MAA.md.2	事前条件、事後条件、不変表明
		MAA.md.3	数理モデルと仕様記述言語(Zや VDM)の紹介
		MAA.md.4	モデリング言語の性質
		MAA.md.5	モデルの記述と意味(モデルの表現の理解)
		MAA.md.6	明快性(前提が全くない場合、全ての前提を記述する場合)
ソフトウェアプロセスと品質			
FND			数理基礎・工学基礎
	FND.ec		ソフトウェアのためのエンジニアリングエコノミクス
		FND.ec.1	ソフトウェアライフサイクルを通じた価値の考慮
		FND.ec.2	システム目的の策定(ユーザ参加型設計、ステークホルダー間のWin-Win関係、品質機能展開、プロトタイピングなど)
		FND.ec.3	費用対効果の評価(利益実現、トレードオフ分析、コスト分析、ROI分析など)
		FND.ec.4	システム価値の実現(優先順位付け、リスクの解決、コストのコントロールなど)
PRF			プロフェッショナルプラクティス
	PRF.pr		プロフェッショナリズム
		PRF.pr.6	ソフトウェアの経済的重要性
		PRF.pr.7	雇用形態・雇用契約
EVO			ソフトウェアの進化や保守
	EVO.pro		進化や保守のプロセス
		EVO.pro.1	進化や保守の基礎
		EVO.pro.2	進化や保守の対象間との関係(前提、要求、アーキテクチャ、設計、ソースコードなど)
		EVO.pro.3	進化や保守のモデル(理論面、法制面など)
		EVO.pro.4	進化や保守のコストモデル
		EVO.pro.5	進化や保守の計画(アウトソーシング、内製など)
	EVO.ac		進化や保守のアクティビティ
		EVO.ac.1	レガシーシステムの扱い(ラッパーの利用など)
		EVO.ac.2	プログラムの理解とリバースエンジニアリング
		EVO.ac.3	(技術面とビジネス面での)システムおよび業務プロセスのリエンジニアリング
		EVO.ac.4	影響解析
		EVO.ac.5	(技術面とビジネス面での)マイグレーション(移行)
		EVO.ac.6	リファクタリング

		EVO.ac.7	プログラム変換
		EVO.ac.8	データのリバースエンジニアリング
PRO			ソフトウェア開発プロセス
	PRO.con		プロセスの基礎
		PRO.con.1	プロセスの概念と用語
		PRO.con.2	プロセスのインフラストラクチャ(開発者、ツール、教育など)
		PRO.con.3	プロセスのモデリングと仕様化
		PRO.con.4	プロセスの測定と分析
		PRO.con.5	プロセスの改善(個人の改善、チームの改善)
		PRO.con.6	品質の分析とコントロール(欠陥予防、レビュー、品質特性、根本原因分析など)
		PRO.con.7	プロセスモデルのモデリングと分析
	PRO.imp		プロセスの実装
		PRO.imp.1	プロセス定義のレベル(組織、プロジェクト、チーム、個人など)
		PRO.imp.2	ライフサイクルモデル(アジャイル、ヘビーウェイト、ウォーターフォール、スパイラルなど)
		PRO.imp.3	ライフサイクルモデルの標準(IEEE標準、ISO標準など)
		PRO.imp.4	個人によるソフトウェアプロセス(モデル、定義、測定、分析、改善)
		PRO.imp.5	チームによるソフトウェアプロセス(モデル、定義、組織、測定、分析、改善)
		PRO.imp.6	プロセスのテーラリング
		PRO.imp.7	ISO/IEEE 12207: プロセスの要求
QUA			ソフトウェア品質
	QUA.cc		ソフトウェア品質の概念と文化
		QUA.cc.1	品質という概念の定義
		QUA.cc.2	品質に対する社会の関わり方
		QUA.cc.3	低い品質によって発生するコストと影響
		QUA.cc.4	品質モデルのコスト
		QUA.cc.5	ソフトウェアの品質特性(信頼性、ユーザビリティなど)
		QUA.cc.6	品質向上に対する取り組みの次元
		QUA.cc.7	人、プロセス、技法、ツール、技術の役割
	QUA.std		ソフトウェア品質に関する標準
		QUA.std.1	ISO 9000シリーズ
		QUA.std.2	ISO/IEEE 12207: 「包括的」な標準

		QUA.std.3	組織ごとの標準の適用
		QUA.std.4	IEEEにおける品質関連の標準
	QUA.pro		ソフトウェア開発プロセスの改善
		QUA.pro.1	ソフトウェアプロセス改善のモデルとメトリクス
		QUA.pro.2	プロセス改善モデルの持つ品質に関する側面
		QUA.pro.3	ISO15504の概要
		QUA.pro.4	ISO 15504における品質関連の領域
		QUA.pro.5	ソフトウェア CMMおよび CMMIにおける品質関連の領域
		QUA.pro.6	ソフトウェア産業に対するマルコムボルドリッジ賞の基準
		QUA.pro.7	その他のプロセス(改善)モデル
	QUA.pca		プロセスの保証
		QUA.pca.1	プロセス保証の起源
		QUA.pca.2	品質計画
		QUA.pca.3	プロセス保証の整理と報告
		QUA.pca.4	プロセス保証の技法
	QUA.pda		製品の保証
		QUA.pda.1	製品保証の起源
		QUA.pda.2	保証と V&Vの違い
		QUA.pda.3	製品品質モデル
		QUA.pda.4	根本原因分析と欠陥予防
		QUA.pda.5	製品品質のメトリクスと測定
		QUA.pda.6	品質特性のアセスメント(ユーザビリティ、信頼性、アベイラビリティなど)
ヒューマンファクター			
CMP			コンピュータ基礎
	CMP.cf		コンピュータ科学基礎
		CMP.cf.7	ヒューマンファクターの基礎(ユーザ側:入出力、エラーメッセージ、障害対応)
		CMP.cf.8	ヒューマンファクターの基礎(開発者側:コメント、構造、可読性)
DES			ソフトウェア設計
	DES.hci		ヒューマン・コンピュータ・インタフェース(HCI)設計
		DES.hci.1	一般的な HCI設計の原則
		DES.hci.2	モードやナビゲーションの使用
		DES.hci.3	コード化の技法とビジュアルデザイン(色、アイコン、フォントなど)
		DES.hci.4	応答時間とフィードバック

		DES.hci.5	デザインのリモダリティ(メニュー駆動、フォーム、問題回答型など)
		DES.hci.6	ローカライゼーションと国際化
		DES.hci.7	HCI設計の技法
		DES.hci.8	マルチメディア(入出力技術、音声認識、自然言語認識、Web、音響効果など)
		DES.hci.9	メタファとコンセプトモデル
		DES.hci.10	HCIの心理学
VAV			検証と妥当性確認(V&V)
	VAV.hct		HCIのテストと評価
		VAV.hct.1	usefulnessとユーザビリティの側面の多様性
		VAV.hct.2	経験則による評価
		VAV.hct.3	認知的ウォークスルー
		VAV.hct.4	ユーザによるテストのためのアプローチ(観察セッションなど)
		VAV.hct.5	Webサイトや Webアプリケーションにおけるユーザビリティおよびテスト技法
開発マネジメント			
FND			数理基礎・工学基礎
	FND.ec		ソフトウェアのためのエンジニアリングエコノミクス
		FND.ec.2	システム目的の策定(ユーザ参加型設計、ステークホルダー間のWin-Win関係、品質機能展開、プロトタイピングなど)
PRF			プロフェッショナルプラクティス
	PRF.psy		グループダイナミクス / 心理学
		PRF.psy.1	チームやグループでの作業の際のダイナミクス
		PRF.psy.2	個人の認知(制限など)
		PRF.psy.3	認知的問題の複雑性
		PRF.psy.4	ステークホルダーとの対話
		PRF.psy.5	不確実性と曖昧性の取り扱い
		PRF.psy.6	多国籍・多文化環境の取り扱い
	PRF.com		(SEに特化した)コミュニケーションスキル
		PRF.com.1	読解、理解、要約(ソースコードやドキュメントなど)
		PRF.com.2	記述(職務記述書、報告書、評価報告書、理由書など)
		PRF.com.3	チームとグループのコミュニケーション(口頭、文書、電子メールなど)
		PRF.com.4	プレゼンテーションスキル
	PRF.pr		プロフェッショナルリズム
		PRF.pr.1	ア krediyetasyon、資格認定、免許制度

		PRF.pr.2	倫理綱領とプロフェッショナルとしての行動
		PRF.pr.3	社会的、法的、歴史のおよびプロフェッショナルとしての考慮事項
		PRF.pr.4	プロフェッショナル・ソサエティ(学会や協会、コミュニティなど)の起源と役割
		PRF.pr.5	標準の起源と役割
MAA			ソフトウェアのモデリングと分析
	MAA.rfd		要求分析の基礎
		MAA.rfd.1	要求という概念の定義(プロダクト、プロジェクト、制約、システムの境界、システムの外部、システムの内部など)
		MAA.rfd.2	要求分析のプロセス
		MAA.rfd.3	要求のレベル/階層(ニーズ、ゴール、ユーザ要求、システム要求、ソフトウェア要求など)
		MAA.rfd.4	要求が備えるべき特性(検証可能性、非曖昧性、一貫性、正当性、トレーサビリティ、優先度など)
		MAA.rfd.5	品質特性(非機能特性)の分析(安全性、セキュリティ、ユーザビリティ、性能など)
		MAA.rfd.6	優先順位付け、トレードオフ分析、要求のためのリスク分析
		MAA.rfd.7	要求とアーキテクチャの競合
		MAA.rfd.8	システムエンジニアリングや人間中心設計と要求との関係
		MAA.rfd.9	厄介型問題(構造の歪みに起因する問題、解がたくさんある問題など)
		MAA.rfd.10	制約としての COTS
	MAA.er		要求の獲得
		MAA.er.1	要求を獲得する対象(ステークホルダー、ドメインエキスパート、操作環境や組織環境など)
		MAA.er.2	要求獲得の技法(インタビュー、アンケート/調査、プロトタイプング、ユースケース、観察、参加型技法など)
		MAA.er.3	進んだ要求獲得の技法(民族学的技法、知識獲得など)
	MAA.rsd		要求の仕様化と文書化
		MAA.rsd.1	要求の文書化の基礎(種類、読み手、構造、品質、属性、標準など)
		MAA.rsd.2	ソフトウェア要求の仕様化
		MAA.rsd.3	仕様記述言語(構造化自然言語、UML、Z・VDM・SCR・RSMLなどの形式言語)
MGT			ソフトウェア開発のマネジメント
	MGT.con		マネジメントの基礎
		MGT.con.1	一般的なプロジェクトマネジメント

		MGT.con.2	古典的なマネジメントモデル
		MGT.con.3	プロジェクトマネジメントの役割
		MGT.con.4	全社的/組織的なマネジメントの構造
		MGT.con.5	ソフトウェア開発マネジメントの種類(調達マネジメント、プロジェクトマネジメント、開発マネジメント、保守マネジメント、リスクマネジメントなど)
	MGT.pp		プロジェクトの計画
		MGT.pp.1	計画と評価
		MGT.pp.2	WBS(Work Breakdown Structure)の作成
		MGT.pp.3	タスクのスケジューリング
		MGT.pp.4	工数の見積り
		MGT.pp.5	リソースの割り当て
		MGT.pp.6	リスクマネジメント
	MGT.per		プロジェクトのメンバと組織
		MGT.per.1	組織構造、地位、責任、権限
		MGT.per.2	フォーマルおよびインフォーマルなコミュニケーション
		MGT.per.3	プロジェクト要員
		MGT.per.4	メンバの教育、キャリア開発、評価
		MGT.per.5	会議のマネジメント
		MGT.per.6	チームビルディングとモチベーション向上
		MGT.per.7	対立の解消
	MGT.ctl		プロジェクトのコントロール
		MGT.ctl.1	変更のコントロール
		MGT.ctl.2	モニタリングと報告
		MGT.ctl.3	結果の測定と分析
		MGT.ctl.4	是正とリカバリ
		MGT.ctl.5	報酬と規律
		MGT.ctl.6	業務遂行能力の基準
	MGT.cm		ソフトウェア構成管理
		MGT.cm.1	リビジョン管理
		MGT.cm.2	リリース管理
		MGT.cm.3	支援ツール
		MGT.cm.4	ビルド
		MGT.cm.5	構成管理のプロセス
		MGT.cm.6	保守に関する考慮事項
		MGT.cm.7	分散環境とバックアップ

付録2 J07-SE の情報科学基礎科目

授業科目名	コンピュータとソフトウェアの基礎	
単位数	2	
開設学期	1年生前期	
目的	コンピュータの構成と動作原理やシステムの基本的な概念、さらに、コンピュータ上で動作するソフトウェアおよびその開発技術の概念と意義について理解することを目的とする。	
概要	コンピュータの概念、基本構造、データのメモリ上での表現、アセンブリ言語の構成、メモリシステム、入出力インタフェースといった一連のコンピュータの構成とシステムの基本的な概念、ならびにソフトウェア開発の全体像を学ぶ。	
目標	コンピュータの構成と動作原理、システムの基本的な概念、ソフトウェアおよびその開発技術の基本的な概念を習得する	
授業展開		学習成果
1	SEコースを概説し、工学的アプローチについて述べる。また本講義の概要と目的および進め方について説明する(※講義0.3h)。	
2	コンピュータの概念と基本的な構造を説明する。	CMP.cf.5
3	コンピュータの命令語の構成と種類、割り込みについて説明する。	CMP.cf.5
4	2進数とN進数、数の計算機上での表現、各種文字コード、データの表現方法について説明する。	CMP.cf.5
5	メモリ階層について紹介し、仮想記憶方式とキャッシュの仕組みについて説明する。	CMP.cf.5
6	入出力、割り込み、インタフェース、ネットワーク技術とネットワーク階層について説明する。	CMP.cf.5
7	プログラムの概念と定義、特長、言語処理系、各種のプログラム言語について説明する。	CMP.cf.5
8	プログラムの基本となる制御構造、アルゴリズムの記法、構造化プログラムについて説明する。	CMP.cf.5
9	構造化プログラム、プログラムとソフトウェアについて説明する。	CMP.cf.5
10	ソフトウェア工学の背景、定義、成果について説明する。	
11	ソフトウェア開発プロセスとプロセスモデルについて説明する。	
12	開発プロジェクトとプロジェクト管理、要員、ソフトウェア品質、コスト、納期について説明する。	
13	システムエンジニアリングと各種システムについて説明する。	CMP.cf.6

14	重要事項のまとめ	
15	期末試験	
教科書・ 参考書	ロジャー・プレスマン: 実践ソフトウェアエンジニアリング, 日科技連, 2005. エリック・プロディ: ソフトウェアエンジニアリング, 翔泳社, 2004	
備考		

授業科目名	確率統計	
単位数	2	
開設学期	2年生前期	
目的	確率モデルや統計分析手法の実装を必要とするソフトウェアの開発に必要な, およびソフトウェア開発プロジェクトを通じて測定されたデータの分析に必要な確率統計の基礎知識の習得を目的とする。	
概要	科学技術分野などのコンピュータシステムでは, 確率モデルや統計分析手法のソフトウェアによる実現が求められる場合が少なくない。また, ソフトウェア工学の観点からは, ソフトウェア開発プロジェクトを定量的に管理することが, 品質・コスト・納期の問題解消に貢献する。したがって, ソフトウェア開発およびその管理の両面において, 確率統計の基礎知識は必須である。以上の問題意識に基づいて, 本講義では, 確率統計および統計解析の基礎知識を学ぶ。	
目標	確率統計の基礎知識を理解し, 事象の確率計算, データの適切な表現, 回帰分析や検定など基本的な統計解析が行えることを目標とする。	
授業展開		学習成果
1	授業の概要と目的および進め方について説明する(※講義0.3h)。記述統計と推測統計の違い, 母集団と標本の意味, 変数の種類, 一変数データのグラフ表現における留意点について説明する。	FND.ef.2
2	一変数データの中心とばらつきを記述する基本的な統計量と, データの分布に用いられる箱ひげ図について説明する。	FND.ef.2
	有効数字の計算における留意点を学ぶ。	FND.mf.9
3	二変数データを記述する方法として, 相関と散布図について説明する。また, 最小二乗法により回帰直線を得る方法を説明し, モデル式を用いた予測が行えることを説明する。	FND.ef.2
4	確率の基本概念について説明する。	FND.mf.6
5	離散確率変数と連続確率変数の違いを説明し, 離散確率変数の確率分布について説明する。	FND.mf.6
6	二項分布について説明し, 二項確率を数式により求める方法と, 累積二項確率分布表から求める方法を説明する。	FND.mf.6

7	正規分布について説明し、正規分布に従ったデータを標準化して正規分布表から確率を求める方法を説明する。	FND.ef.2
	測定値と誤差の関係を説明し、多くの測定値が正規分布に従うことを説明する。	FND.mf.9
8	サンプリングの方法を説明し、母集団分布と標本分布の違いについて説明する。また、中心極限定理について説明する。	FND.ef.2
9	標本平均および標本比率の標本分布における平均と標準誤差について説明する。また、統計的工程管理（SPC）の基礎について説明する。	FND.ef.2
10	点推定と区間推定について説明し、許容誤差を与えることで必要なサンプルサイズを決定できることを説明する。	FND.ef.2
11	統計的仮説検定の基本的な考え方を説明し、大標本における統計的仮説検定（母分散が既知の場合）の方法を説明する。	FND.ef.2
12	大標本における統計的仮説検定について、二群の母平均の差の検定と、母比率の検定について説明する。また、信頼区間と検定の関係性について説明する。	FND.ef.2
13	小標本における統計的仮説検定について、検定の対象によって異なる分布を用いることを説明する。	FND.ef.2
14	重要事項のまとめ	
15	期末試験	
教科書・参考書	Mendenhall, W., Beaver, R.J. and Beaver, B.M., Introduction to Probability and Statistics, 12th ed., Thomson, 2006. 東京大学教養学部統計学教室(編), 統計学入門, 東京大学出版会, 1991.	
備考		

授業科目名	離散数学	
単位数	2	
開設学期	2年生後期	
目的	ソフトウェア工学において必要になる離散数学の様々な分野について学習することを目的とする。	
概要	グラフ理論、代数学、数論、数え上げの基礎について学ぶ。	
目標	ソフトウェア工学の理解に必要な離散数学の知識を獲得する。	
授業展開		学習成果

1	授業の概要と目的および進め方について説明する(※講義0.3h)。 理解に必要な、集合論の基礎を説明してから、基本的な諸概念を、実例などを交えて説明をする。	FND.mf.5
2	様々なグラフの中から重要な応用(例えば、オイラー路)について教える。	FND.mf.5
3	グラフの探索アルゴリズムとその実装について説明する。	FND.mf.5
4	群に関する基礎的な諸概念を説明する。	FND.mf.11
5	環に関する代数的基礎について説明をする。	FND.mf.11
6	体に関する代数的基礎について説明をする。	FND.mf.11
7	計算機科学において広く用いられている、順序集合とそれ上の代数構造である束について学び、ブール代数まで拡張した代数系の説明を行う。	FND.mf.11
8	数(自然数、整数、素数)に関する基礎知識を説明する。	FND.mf.10
9	整数に関する基本的定理について説明を行う。	FND.mf.10
10	整数論の暗号への応用について説明を行う。	FND.mf.10
11	数え上げの基礎概念について説明をする。	FND.mf.4
12	順列と組み合わせ理論についてより複雑な概念といくつかの結果について説明を行う。	FND.mf.4
13	組み合わせ理論において重要なカタラン数、スターリング数、フィボナッチ数などについて説明をする。	FND.mf.4
14	重要事項のまとめ	
15	期末試験	
教科書・参考書	やさしい組み合わせ数学、西岡弘明、コロナ社(1999)、グラフ・ネットワーク・組み合わせ論、藤重悟、共立出版(2002)	
備考		

授業科目名	プログラミング基礎
単位数	2
開設学期	2年生後期
目的	プログラミング言語によりソフトウェアをどのように作成するかを理解し、そのうえで効率よく問題を解決するためのアルゴリズムについて基礎を学ぶ

概要	C言語を対象にして、プログラミング言語によりソフトウェアをどのように作成するのかという基本を理解する。さらに、問題を解決するためのアルゴリズムの概念と実装の基礎について理解する	
目標	プログラムの作成方法や実行形式への変換方法を学ぶ。変数、データ型、演算、分岐、繰り返し、配列操作などの基本的なプログラムの文法を理解し、これらを用いてプログラムが作成できるようになる。	
授業展開		学習成果
1	講義を通して学ぶプログラミング言語であるC言語の概要及び発展の歴史を学び、さらにプログラミング環境の基本的な使い方を習得する	CMP.cf.9
2	変数の宣言方法、データ型、演算子などを学び、それらを使ってプログラムを作成する	CMP.cf.1
3	分岐と繰り返しの2種類の制御文を学ぶ	CMP.cf.1
4	1次元配列や多次元配列の操作方法を学ぶ	CMP.cf.1
5	関数の定義方法を学ぶ	CMP.cf.1
6	文字列操作、ポインター、再起呼び出しを学ぶ	CMP.cf.1
7	イベント駆動型プログラミングとプログラムの品質や信頼性を高める方法を学ぶ	CMP.cf.1,4
8	プログラミング環境とツールを学ぶ	CMP.tl.1,2,3,5
9	アルゴリズムの基礎	CMP.cf.1, CMP.cf.2
10	アルゴリズム設計手法	CMP.cf.1,2
11	基礎的なアルゴリズム	CMP.cf.2
12	探索アルゴリズムとデータ構造	CMP.cf.1, CMP.cf.2
13	分散アルゴリズムと計算可能性	CMP.cf.2
14	重要事項のまとめ	
15	期末試験	
教科書・参考書	C言語の場合：柴田望洋，明解C言語入門編，ソフトバンククリエイティブ，2004 浅野，増沢 利光，和田 幸一，アルゴリズム論（IT Text）（単行本），オーム社，2003 計算とアルゴリズム（新コンピュータサイエンス講座）：浅野孝夫，今井弘，オーム社，2000	
備考	本カリキュラムは、プログラミング初心者を対象とした弊社の新人研修の講義及び実習内容と、柴田望洋さんの「新版明解C言語入門編」を参考に作成しました	

授業科目名	論理と推論・計算理論	
単位数	2	
開設学期	2年前期	
目的	本講義では、論理学と計算理論の基礎を学ぶことを目的とする。	
概要	ソフトウェアを形式的に特徴づけたり解析するためには、計算や論理に関する知識が不可欠である。本講義では、論理学や計算理論を中心に、関連する議論も含めて学習する。	
目標	<ul style="list-style-type: none"> ・集合、関係、関数に関する基本的な知識を習得する。 ・命題論理および述語論理に関する基本的な知識を習得する。 ・言語理論と計算理論に関する基本的な知識を習得する。 	
授業展開		学習成果
1	授業の概要と目的および進め方について説明する。集合の表現法と意味、および、基本的な演算について説明する。	FND.mf.1
2	集合の諸概念(有限集合、無限集合、濃度など)について説明する。	FND.mf.1
3	集合に基づいた関係とその分類について説明する。関数、および、その分類について説明する。	FND.mf.1
4	命題論理の論理式と諸概念(真理値表、トートロジー、同値性、標準形など)について説明する。	FND.mf.2
5	命題論理の形式的体系とそれによる証明について説明する。形式的体系の性質について説明する。	FND.mf.3
6	述語論理の論理式と諸概念(述語変数、量化記号、束縛変数、自由変数など)について説明する。	FND.mf.2
7	構造と述語論理の意味について説明する。	FND.mf.2
8	述語論理の諸概念(恒真性、充足可能性、冠頭標準形など)について説明する。	FND.mf.2
9	述語論理の形式的体系とそれによる証明について説明する。形式的体系の性質について説明する。命題論理・述語論理以外の論理があることを説明する(様相論理など)。	FND.mf.3
10	有限オートマトンの概要について説明する。有限オートマトンの定義と、その表現能力について説明する。	FND.mf.7
11	決定性オートマトン、非決定性オートマトンの定義と諸概念(等価性など)について説明する。計算量(P、NP、PSPACE)について説明する。	FND.mf.7
12	正規表現の定義と諸概念(有限オートマトンとの等価性など)について説明する。	FND.mf.7
13	文脈自由文法と諸概念について説明する。	FND.mf.8

14	重要事項のまとめ	
15	期末試験	
教科書・参考書	田辺誠・中島玲二・長谷川真人,「コンピュータサイエンス入門」, 岩波書店 小野寛晰,「情報科学における論理」, 日本評論社 J. ホップクロフト・J. ウルマン・R. モトワニ(野崎・町田・高橋・山崎訳)「オートマトン 言語理論 計算論(第2版)」, サイエンス社	
備考		

授業科目名	オペレーティングシステム基礎・データベース基礎	
単位数	1	
開設学期	2年生前期	
目的	オペレーティングシステムの基本的な役割、データベースのモデル、言語、設計手法、運用管理手法を学習することを目的とする	
概要	オペレーティングシステムは、プログラマがハードウェアを制御できるよう抽象化し、ユーザの間で共有する資源を管理する。このエリアのトピックは、現在のオペレーティングシステムの設計に影響を与える課題について、取り扱う。 また、大量のデータを効率よく利用するために欠かすことのできないデータベースに関して取り上げ、データベースのモデル、言語、設計手法、運用管理手法を学習する。	
目標	オペレーティングシステムの役割と基本的概念を把握し、コンピュータシステムの働きを理解するとともに、各種基本構成要素とその働きとについて理解することを目的とする。 データベースの歴史から組織での必要性を理解し、データベース言語SQLが使用できること、論理データモデルのモデル化方法と物理データベース設計方法を用いてデータベース設計ができることを目標とする。また、データベース管理システムについての知識を持ち、運用管理の方法を理解していることを目標とする	
授業展開		学習成果
1	授業の概要と目的および進め方について説明する(※講義0.3h)。 オペレーティングシステムの概念や位置づけを説明する。	CMP.cf.10
2	オペレーティングシステムの仕組みについて説明する。 プロセスの管理について説明する。(1)	CMP.cf.10
3	プロセスの管理について説明する。(2) プロセスのスケジューリングについて説明する。	CMP.cf.10
4	プロセスの同期について説明する。	CMP.cf.10
5	メモリ管理について説明する。 仮想記憶について説明する。(1)	CMP.cf.10
6	仮想記憶について説明する。(2)	CMP.cf.10

	入出力制御と割込みについて説明する。	
7	ファイルシステムについて説明する。	CMP.cf.10
8	データベースに関連する基本的な用語や概念を説明する	CMP.cf.11
9	データベース言語に関して説明する	CMP.cf.11
	具体的例題を与えて、SQLの記述演習を行う	
10	データベース設計の概要と論理設計を説明する	CMP.cf.11
	具体的例題を与えて、論理設計の演習を行う	
11	物理設計に関して説明する	CMP.cf.11
12	データベースシステムの構成と障害回復に関して説明する	CMP.cf.11
13	データベースの周辺技術に関して説明する	CMP.cf.11
14	重要事項のまとめ	
15	期末試験	
教科書・参考書	「オペレーティングシステムの基礎」 大久保英嗣 著 (サイエンス社) ISBN 4-7819-0860-8 「オペレーティングシステム」 清水謙多郎 著 (岩波書店) ISBN 4-00-007852-6 「オペレーティングシステム」 野口健一郎 著 (オーム社) ISBN 4-274-13520-1 「リレーショナルデータベース入門」 増永良文 著 サイエンス社 ISBN 4-781-91024-6 「データベースシステム」 北川博之 著 昭晃堂 ISBN 4-785-62046-3	
備考		

授業科目名	ネットワーク基礎	
単位数	2	
開設学期	2年後期	
目的	情報ネットワーク構成の基礎知識、具体的な事例に関する知識の獲得により、様々なアプリケーションにおいて、適切な情報通信技術を選択し利用することができる能力を身につけることを目的とする。	
概要	情報ネットワークを構成する様々な技術要素について基礎的な側面から具体的な応用例までを総合的に学ぶ。ネットワークを構成する各レイヤ技術についても学習する。	
目標		
授業展開		学習成果

1	ネットワーク・インターネットの背景・歴史について説明する	CMP.cf.12
	普及しているネットワークアプリケーション事例について説明する	
	ネットワークの基礎概念について説明する	
2	OSI参照モデルの概要とTCP/IPの位置づけについて説明する	CMP.cf.12
	OSI参照モデルの物理層について説明する	
	OSI参照モデルのデータリンク層について説明する	
3	OSI参照モデルのネットワーク層とIPについて説明する	CMP.cf.12
	IPに関連したプロトコルについて説明する	
	ルーティングについて説明する	
4	OSI参照モデルのトランスポート層とTCP/UDPの位置づけについて説明する	CMP.cf.12
	UDPについて説明する	
	TCPについて説明する	
5	TCP/IPを用いたプログラミングについて説明する	CMP.cf.12
	TCP/IPを用いたプログラミング演習を行う	
6	OSI参照モデルのセッション層, プレゼンテーション層, アプリケーション層について説明する	CMP.cf.12
	様々なアプリケーションプロトコルの概要について説明する	
7	Webブラウザ・Webサーバの基本的な仕組みについて説明する	CMP.cf.12
	Webにおいて利用されるプログラムの実行方式について説明する	
8	HTTPを例としてクライアントサーバプログラミングについて説明する	CMP.cf.12
	HTTPを例としてクライアントサーバプログラミングの演習を行う	
9	HTTPの詳細や, FTPやPOP3等他のプロトコルを例として, クライアントサーバプログラミングについて説明する	CMP.cf.12
	HTTPの詳細や, FTPやPOP3等他のプロトコルを例として, クライアントサーバプログラミングの演習を行う	
10	セキュリティの基本概念について説明する	CMP.cf.12
	暗号・認証の仕組みについて説明する	
	暗号・認証の運用について説明する	
11	システムセキュリティについて説明する	CMP.cf.12
	ネットワークセキュリティについて説明する	
12	セキュリティ運用について説明する	CMP.cf.12
	セキュリティに関する事例について説明, 議論する	
13	関連技術や実用ツールについて説明する	CMP.cf.12

	最新のネットワーク技術について説明する	
14	重要事項のまとめ	
15	期末試験	
教科書・参考書	「マスタリングTCP/IP 入門編」(竹下隆史・村山公保・荒井 透・苅田幸雄, オーム社), 「マスタリングTCP/IP 応用編」(Philip Miller著, 苅田 幸雄監訳, オーム社), 「コンピュータネットワーク」(Andrew S. Tanenbaum 著, 日経BP社), 「情報セキュリティ技術大全 : 信頼できる分散システム構築のために」(Ross Anderson著, トップスタジオ訳, 日経BP社)	
備考		

授業科目名	工学基礎	
単位数	2	
開設学期	4年生前期	
目的	社会に出て活躍するために専門性の高い技術に加え、一般的な工学の概念を身に付け、物事を論理的に深く洞察し問題に対しては解決手法を提示できるようになることを目的とする	
概要	問題解決技法、統計解析、測定とメトリクス、システム特性、工学的設計の基本概念、測定の理論、倫理綱領とプロフェッショナルとしての行動姿勢を学ぶほか、工学原理についても触れる	
目標	問題解決の技法を知り、事象をシステムティックに捉え、問題を定式化し、解決の手法を見出し、分析、評価して解決することができる	
授業展開		学習成果
1	授業の概要と目的、進め方、評価基準について説明する(0.3h) 工学的設計の基本概念について解説する(1)(講義冒頭0.1hで本日の講義の概要を説明)	FND.ef.5,7
2	工学的設計の基本概念について解説する(2)(講義冒頭0.1hで本日の講義の概要を説明)	MAA.af.1,2,3,4,5
3	工学的設計の基本概念について解説する(3)(講義冒頭0.1hで本日の講義の概要を説明)	FND.ef.5
4	問題解決技法とは何か、代表的な問題解決サイクルについて解説する(講義冒頭0.1hで本日の講義の概要を説明)	CMP.cf.3
5	ソフトウェア開発と問題解決技法との関係を説明する	CMP.cf.3
6	システムの基礎(システムとは何か、システムにはどのようなものがあるか)について解説する(講義冒頭0.1hで本日の講義の概要を説明)	CMP.cf.6
7	統計的技法と問題解決技法について解説する(講義冒頭0.1hで本日の講義の概要を説明)	FND.ef.1
8	統計的技法の使い方について解説する(講義冒頭0.1hで本日の講義の概要を説明)	FND.ef.6

9	測定理論(講義冒頭0.1hで本日の講義の概要を説明)	FND.ef.6
10	ソフトウェアの品質、生産性を測定する指標について解説する(講義冒頭0.1hで本日の講義の概要を説明)	FND.ef.3
11	ISOで定められているシステム特性について解説する(講義冒頭0.1hで本日の講義の概要を説明)	FND.ef.4
12	ソフトウェア工学以外の分野における工学的な原理について紹介する、サイエンスと工学の関係を説明する(講義冒頭0.1hで本日の講義の概要を説明)	FND.ef.7
13	技術者としての倫理綱領とプロフェッショナルとしての行動指針について解説する(講義冒頭0.1hで本日の講義の概要を説明)	PRF.pr.2
14	重要事項のまとめ	
15	期末試験	
教科書・参考書	<ul style="list-style-type: none"> ・仙場洋一 著 「信頼を得る 上級SEの問題解決力-SEが直面する33の問題シーン-」 (技術評論社、ISBN 4-7741-1817-6) ・宮川宏之 著 「どこでどう使うQC手法」 (ブレーン・ダイナミクス、978-4-938340-28-5 (4-938340-28-3)) ・ローレンス・H.パトナム/著 ウエア・マイヤーズ/著 山浦恒央/訳 「初めて学ぶソフトウェアメトリクス」 (日経BP社、978-4-8222-8242-4 (4-8222-8242-2)) ・ロジャー・プレスマン: 実践ソフトウェアエンジニアリング, 日科技連, 2005. ・エリック・ブロディ: ソフトウェアエンジニアリング, 翔泳社, 2004 ・Frye, Ellen., Engineering Problem Solving for Mathematics, Science, and Technology Education (http://engineering.dartmouth.edu/teps/book.html)に目次あり。 ・Dym, C. L., and Little, M. (2000): Engineering Design: A Project-Based Introduction, John Wiley & Sons, New York 	
備考		

付録3 J07-SEのSE科目

授業科目名	ソフトウェア構築	
単位数	2	
開設学期		
目的	ソフトウェアを構築するにあたり必要とされる過程を理解し、構築の際に用いられる技術について学習することを目的とする。	
概要	ソフトウェアの構築とは何を行うことかを習得する。ソフトウェアを構築するにあたり、必要とされる過程とその重要性、および構築のための要素技術を習得する。	
目標	作成されるソフトウェアの設計内容が与えられた際、自分で、あるいは多人数で共同してソフトウェアを構築する能力を身につける。ソフトウェアを構築する他の作業者に、ソフトウェアを構築するために必要な作業が何かを指示し管理する能力を身につける。	
授業展開		学習成果
1	授業の概要と目的および進め方について説明する。	
	ソフトウェア開発の全体的な流れについて説明する。また、その流れの中でソフトウェア構築がどのような位置づけであるか説明する。	PRO.con
	ソフトウェア開発の要素としてどのような作業があるかを理解する。	PRO.con
2	ソフトウェア開発の全体的な流れの中で、ソフトウェア構築がどのような位置づけであるか、ソフトウェア設計との関連から説明する。	DES.con
3	ソフトウェア構築における開発環境の役割について説明する。	CMP.tl.1
4	ソフトウェア設計時に用いるプログラミング言語としてオブジェクト指向言語を取り上げ、その特徴について説明する。	CMP.ct.3,4,5,6
5	GUI構築ツールを用いることにより、ソフトウェア構築をどのように行うことが出来るかを説明する。	CMP.tl.2
	単体テストツールを用いることにより、ソフトウェア構築をどのように行うことが出来るかを説明する。	CMP.tl.3,16
6	複数の実行が並列・並行に動作するソフトウェアの動作原理について説明する。また、そのようなソフトウェアを構築するための技術について説明する。	CMP.ct.10,12
7	ソフトウェア設計の際に状態やテーブルが用いられた際、それをソフトウェア構築時どのようにして実装することが出来るかを説明する。	CMP.ct.7
8	ソフトウェアの振舞いをその実行時に決定する方法について説明する。また、その一例としてソフトウェアの国際化を取り上げ、国際化がどのようにして行われるかを説明する。	CMP.ct.8
9	文字列等を入力とするソフトウェアを対象とし、入力内容をソフトウェア内部で	CMP.ct.9

	利用するために必要となる処理について説明する.	
10	実世界のソフトウェアにおける特定のドメインに関するソフトウェアについて説明する.	CMP.ct.11,13
11	ソフトウェアを構築する段階における, ソフトウェア分析手法について説明する.	CMP.ct.14,CM P.tl.5
12	ソフトウェアを構築する際, 既存のソフトウェアを流用することによって効率的にソフトウェアを作成する方法について説明する. また, 既存のソフトウェアを流用することによる効果について説明する.	CMP.ct.15,1,2
13	ソフトウェア構築をより抽象度の高いレベルで構築するとき用いられるプログラミング言語について説明する.	CMP.tl.4
14	重要事項のまとめ	
15	期末試験	
教科書・参考書		
備考		

授業科目名	モデル化と要求開発	
単位数	2	
開設学期	2年生後期	
目的	ソフトウェアのモデル化と分析のための概念を理解し、良いモデルを構成する能力と良い要求仕様を構成する能力を獲得することを目的とする	
概要	ソフトウェア開発においては、開発対象のモデル化と要求仕様の作成が必須であり、これらの製品の質がソフトウェア開発の成否に関わってくる。本講義では、モデル化の原則、モデルの種類、モデルの分析基礎、要求分析の基礎、要求の獲得、要求の仕様化、要求の妥当性確認といった一連のモデル化技法と要求定義技法について学ぶ。	
目標	ソフトウェア開発者として必要な能力であるソフトウェアモデル化技法と要求定義技法を習得する。具体的には各種モデルを種々の観点から分析できる能力、要求を獲得し高品質な要求を仕様化する能力、要求の妥当性を確認する能力の習得を目標とする	
授業展開		学習成果
1	授業の概要と目的および進め方について説明する(※講義0.3h)。 モデルの概念と意義、モデル化の原則について説明する。	MAA.md.1
2	要求工学について概説し、要求分析のプロセス、要求のレベルについて説明する	MAA.rfd.1,2,3
3	要求仕様の満たすべき品質特性とその分析について述べる	MAA.rfd.4,5,6, 7
4	要求仕様の評価法と構成法について述べる	MAA.rv.4,1,2

5	レビューとインスペクションについて述べる。また要求仕様の妥当性を確認する手法としてプロトタイピングについて述べる	MAA.rv.1,2,3
6	要求獲得の対象と獲得技法の中で問題分析手法とインタラクション分析法、合意形成法について述べる	MAA.er.1,2,8
7	ゴール指向分析と要求獲得プロセス全般を扱う技法を紹介する	MAA.er.3
8	要求をモデル化・仕様化するための言語(要求言語)について述べる	MAA.rsd.3
9	情報やデータ、システムの静的な構造のモデル化について述べる(1)	MAA.tm.1
10	情報やデータ、システムの静的な構造のモデル化について述べる(2)	MAA.tm.3,5
11	対象の動的な振舞いのモデル化について述べる(1)	MAA.tm.2
12	対象の動的な振舞いのモデル化について述べる(2)	MAA.tm.2
13	ドメイン、エンタープライズシステム、組み込みシステムのモデル化手法について述べる	MAA.tm.4,6,7
14	アナリシスパターン、まとめ	MAA.tm.9
15	期末試験	
教科書・参考書	要求工学(大西・郷著)、共立出版	
備考		

授業科目名	ソフトウェアアーキテクチャ	
単位数	2	
開設学期	3年生前期	
目的	設計の概念、設計にて考慮すべき事柄(品質特性、技法、ライフサイクル等)と代表的なアーキテクチャのパターンについて学習することを目的とする。	
概要	ソフトウェアの設計とは何かを習得する。ソフトウェア設計において、考慮すべき品質特性、設計技術、戦略、設計の表現形態、支援ツール、設計の評価方法について習得する。	
目標	設計に用いられる概念、設計のパラダイム、アーキテクチャ設計のスタイル、設計の支援ツールと評価方法について、基本概念を習得し、実際の設計課題に直面した際に、適切な解決策を導き出すための基礎能力を身につける。	
授業展開		学習成果
1	授業の概要と目的および進め方について説明する。必要に応じ、グループ演習用のグループ分けを発表する。(※講義0.3h)。	DES.con.1,2,3,4,5,6

	設計の概念、基本的な考慮事項、開発ライフサイクルにおける位置づけ、設計の原則、要求・品質との関係を説明する。	
2	設計におけるトレードオフ、アーキテクチャのスタイル、パターンなどの考え方があることを説明する。 ソフトウェア設計で説明した設計手法の振り返り(その1) 設計手法の全体像、機能指向、オブジェクト指向、データ中心、アスペクト指向による設計手法の関係について説明する。 さらに、機能指向による設計手法について説明する。	DES.con.7,DES.con.8,DES.str.1
3	ソフトウェア設計で説明した設計手法の振り返り(その2) オブジェクト指向による設計手法、設計結果の記法について説明する。オブジェクト指向による設計の発展形としてアスペクト指向による設計手法について説明する。	DES.str.2,4,DES.ar.6
4	ソフトウェア設計で説明した設計手法の振り返り(その3) 設計手法(データ構造を中心とした設計)について説明する。 アーキテクチャスタイル(その1) アーキテクチャスタイルの基本概念について説明する。	DES.str.3,DES.ar.1
5	アーキテクチャスタイル(その2) アーキテクチャスタイルの続きと、アーキテクチャで考慮すべき各種特性について説明する。 演習では、事例に基づき、アーキテクチャスタイルの具体的適用方法、各種特性のトレードオフについて議論し、結果を発表する。	DES.ar.1,DES.ar.2
6	アーキテクチャで考慮すべき各種特性およびこれらの間でのトレードオフの前の続きを説明する。また、演習の続きを行う。	DES.ar.2
7	ハードウェアに関連してソフトウェアアーキテクチャで考慮すべき事柄と、要求のトレーサビリティについて説明する。	DES.ar.3,4
8	最適化アーキテクチャ(その1) システム全体を考慮したアーキテクチャ設計の最適化について説明する。アーキテクチャ設計にかかわる要求および設計手法について説明する。	
9	最適化アーキテクチャ(その2)システム全体を考慮したアーキテクチャ設計の最適化についての前回の続きを説明する。アーキテクチャ設計の評価手法について説明する。	

10	大規模システムの開発の場合のアーキテクチャ設計手法について、手法の全体像、大規模システムに対する主な要求、その要求に対応するためのアーキテクチャ設計の考え方について説明する。	
11	システム全体のアーキテクチャと要求のトレーサビリティについてと、ドメイン特化のアーキテクチャおよびプロダクトラインについて説明する。	DES.ar.4
12	設計上の特性とそのメトリクス(その1) 設計上の特性の測定方法(定量的方法、定性的方法)と評価方法について説明する。	DES.ste.2,3
13	設計上の特性とそのメトリクス(その2) 設計上の特性の測定方法と評価方法(前回の続き)、アーキテクチャの記述方法とツールサポートについて説明する。	DES.ste.3,4,D ES.ste.4,DES. ar.6,DES.ste.1
14	重要事項のまとめ(各回に時間を分配し、当日の概要及び手法の全体像等について説明することも可)	
15	期末試験	
教科書・参考書	①岸、野田、深澤著:「ソフトウェアアーキテクチャ」ソフトウェアテクノロジーシリーズ4、アーキテクチャとドメイン指向トラック、共立出版 ②Len Bass, Rick Kazman, Paul Clements 著:Software Architecture in Practice (SEI Series in Software Engineering), Addison-Wesley Pub. 翻訳版「実践ソフトウェアアーキテクチャ」日刊工業新聞社 (※必要に応じ、機能指向、オブジェクト指向、アスペクト指向に関する設計手法の文献も選択する。)	
備考		

授業科目名	ソフトウェア設計	
単位数	2	
開設学期	2年生後期	
目的	ソフトウェア設計の概念及び、開発ライフサイクル内での位置づけを学ぶとともに、いくつかの具体的な設計手法に関して学ぶことを目的とする	
概要	ソフトウェア設計の概念を学んだ後、詳細設計の技法として、機能指向の設計、データ構造を中心とした設計、オブジェクト指向の設計、コンポーネント設計に関して学ぶ。	
目標	ソフトウェア設計の概念と設計時に考慮すべき視点を理解し、それぞれの代表的な設計方法、表記方法を身に着けることで、様々なソフトウェア開発に対応できる能力を身につける。	
授業展開		学習成果
1	設計の概念、基本的な考慮事項、開発ライフサイクルにおける位置づけ、を説明する。	DES.con.1,3,4, 2

2	構造化分析設計法を通して、機能指向の設計に関して説明する。	DES.str.1 ,D ES.dd.1,DES.d d.5
3	構造化設計法と凝集度、結合度によるモジュール評価法を説明する。	DES.str.1,DE S.dd.1,DES.dd .5,DES.con.4
4	データ中心設計に関して説明する	DES.str.3 ,DE S.dd.5,DES.dd .1
5	オブジェクト指向設計法の導入を行う。 ユースケースとロバストネス分析、CRC法を説明する	DES.str.2
6	オブジェクト指向における基本的な概念を説明する UMLとUMLエディタを利用して、静的構造の基本的な表現方法を説明する 例題を利用した、基本的静的構造のUMLモデリング演習を行う	DES.str.2,DE S.dd.5,DES.st e.1,DES.dd.5
7	UMLによる、設計の動的構造の表現方法を説明する	DES.dd.5
8	状態機械図と他の相互作用図の説明を行う 状態機械図による設計表現方法を説明する リアクティブシステムの設計方法に関して説明する	DES.dd.5,1
9	契約による設計法を通して、オブジェクト指向設計における設計原則を説明する。 コンポーネントとシステムのインタフェース設計を説明し、 契約による設計法を説明する	DES.con.4, DES.dd.3, DES.dd.1,3,4
10	既存のコンポーネント技術に関して説明を行う	DES.dd.3
11	構造、生成に関するデザインパターンを説明する	DES.dd.2,5
12	振る舞いに関するデザインパターンを説明する	DES.dd.2,5
13	フレームワークに関して説明する	DES.dd.3
14	重要事項のまとめ	
15	期末試験	
教科書・参考書		
備考		

授業科目名	検証と妥当性確認(V&V)
単位数	2
開設学期	3年生後期

目的	検証と妥当性確認について、その目的と役割を理解し、基本的な事項を説明できることを目的とする。	
概要	検証と妥当性確認では、プログラムが仕様を満足していることや、ステークホルダーの期待を実現していることを保証するために、システムを検査する静的あるいは動的技法の両方を使う。静的技法は、ソフトウェアのライフサイクルのすべての段階を通じて、システム内部表現の解析や検査に用いる。一方、動的技法は、実装されたシステムにのみ適用できる。	
目標	検証と妥当性確認で用いられる用語について、また、各種レビューおよびテスト技法の目的や役割について、それぞれ理解するとともに、検証と妥当性確認の活動を実施するための基礎的事項を習得する。	
授業展開	学習成果	
1	授業の概要と目的および進め方について説明する(※講義0.3h)。 V&Vの概念、基本的な考慮事項、計画や戦略について説明する。	VAV.fnd.1,2,3
2	V&Vの開発ライフサイクルにおける位置づけや、要求や品質との関係について説明する。	VAV.fnd.4,5
3	レビューについて説明する。	VAV.rev.1,2,3
4	単体テストについて説明する。	VAV.tst.1,2
5	ホワイトボックステスト技法について説明する。	VAV.tst.3
6	ブラックボックステスト技法について説明する。(1)	VAV.tst.4
7	ブラックボックステスト技法について説明する。(2)	
8	ソフトウェア結合テストについて説明する。	VAV.tst.5
9	ユースケーステスト技法や運用プロファイルによるテストについて説明する。	VAV.tst.6,7
10	システム総合テストや品質特性に関連するテストについて説明する。	VAV.tst.8,9
11	回帰テストやテストツールについて説明する。	VAV.tst.10,11
12	不具合分析について説明する。(1)	VAV.par.1,2
13	不具合分析について説明する。(2)	VAV.par.3,4
14	重要事項のまとめ	
15	期末試験	
教科書・参考書	「はじめて学ぶソフトウェアのテスト技法」リー・コーブランド 著, 宗雅彦 訳(日経BP社) ISBN 4-8222-8251-1 「JSTQB認定テスト技術者 Foundation Level試験」大西建児, 勝亦匡秀, 加藤大受, 佐々木方規, 鈴木三紀夫, 町田欣史, 吉澤智美, 湯本剛 著(翔泳社) ISBN: 978-4-7981-1204-6	
備考		

授業科目名	形式手法	
単位数	2	
開設学期	3年生前期	
目的	形式手法に関する基礎知識、様々な種類の仕様記述言語、検証方法などについて学ぶ。	
概要	様々な仕様記述言語の概要から、いくつかの仕様記述言語を選択し、その詳細を演習を通じて学ぶ。	
目標	形式手法に関する基礎的な知識を獲得することを目標とする。	
授業展開		学習成果
1	授業の概要と目的および進め方について説明する(※講義0.3h)。 モデリングの原則について説明をする。 モデリング言語一般について説明をする	MAA.md.1, MAA.md.5
2	形式仕様記述言語全般について概説する。	MAA.md.4, MAA.af.6
3	抽象機械について概説する。	MAA.md.3
4	抽象機械を用いて、簡単な例を記述し、演習を行う。	CMP.fm.1 MAA.md.5
5	抽象機械を用いて、より高度な演習を行う。	CMP.fm.1 MAA.md.5
6	形式仕様記述言語(例えばZ)について概要を説明する	MAA.md.3
7	形式仕様記述言語(Z、B、VDM)を用いて、簡単な例を記述し、演習を行う。	CMP.fm.2 MAA.md.5
8	形式仕様記述言語(Z、B、VDM)を用いて、より高度な演習を行う。	CMP.fm.2 MAA.md.5
9	記述の正さと、意味の正しさの違いについて、正当性と妥当性の違いについて説明をする。	CMP.fm.8, CMP.fm.4, CMP.fm.5, CMP.fm.6
10	検証対象のプログラムと検証する性質について説明をする。仕様と記述する性質の関係、記述の明快性を説明する。	MAA.md.2, MAA.md.6, CMP.fm.4, CMP.fm.5, CMP.fm.6
11	代表的なプログラムの検証法。保証できる性質について説明をする。仕様を満	CMP.fm.8,

	たすプログラムについて説明する。	CMP.fm.4, CMP.fm.5, CMP.fm.6
12	簡単な例の検証を行うことで、学生の理解を深める。仕様を満たすプログラムの候補について検討する。	CMP.fm.8, CMP.fm.4, CMP.fm.5, CMP.fm.6
13	仕様から実装を行う手法について説明をする。プログラムを自動生成したり、仕様から詳細化することによりプログラムを導出するという概念を説明する。	CMP.fm.3, CMP.fm.4, CMP.fm.5, CMP.fm.6, CMP.fm.7
14	重要事項のまとめ	
15	期末試験	
教科書・参考書	J. Woodcock, J. Davies, "Using Z", Prentice Hall (1996), 荒木啓二郎 張漢明, 「プログラム仕様記述論」, オーム社(2002) Z. Manna: Mathematical Theory of Computation, McGraw-Hill(1974). E.W.Dijkstra: A discipline of programming, Prentice-Hall(1976). 林晋: プログラム検証論, 共立出版(1995).	
備考		

授業科目名	ソフトウェアプロセスと品質	
単位数	2	
開設学期	3年生前期	
目的	ライフサイクル全体を通じたソフトウェアプロダクトの価値創出のための活動と技術について理解することを目的とする。	
概要	ソフトウェアライフサイクルの主要工程、進化と保守、ソフトウェア品質の概念と管理手法、および、ソフトウェアプロセスの概念とその改善手法を学ぶ。	
目標	ソフトウェアライフサイクルの主要工程とその内容を説明できる。 ソフトウェア進化と保守の目的、作業、技術について説明できる。 ソフトウェア品質の概念や、その管理手法を説明できる。 ソフトウェアプロセスの概念とその改善手法を説明できる。	
授業展開		学習成果
1	授業の概要と目的および進め方について説明する。	FND.ec.1,

	ソフトウェアライフサイクルをシステム価値創出のためのプロセスとして理解し、 価値実現のための技術として品質、プロセス、保守といった概念を位置づけて 説明する	PRO.imp.7,FN D.ec.2,PRF.pr. 7,FND.ec.4
2	ライフサイクルにおける進化や保守の位置づけ	EVO.pro.1,2,3, 4,5
3	進化や保守のための技術1	EVO.ac.1,2,3
4	進化や保守のための技術2	EVO.ac.4,5,6,7 ,8
5	ソフトウェア品質の概念	QUA.cc.1,2,3, 4
6	プロダクト保証とプロセス保証	QUA.cc.6,QU A.pda.1,2,3,4, QUA.pca.1,2,3 ,4
7	ソフトウェアの品質モデルとメトリクス	QUA.pda.5,6, QUA.cc.5,QU A.std.1,QUA.s td.4
8	ソフトウェアプロセスと品質	QUA.cc.7,PR O.com.1.2.3
9	プロセスのモデリング	PRO.imp.1,2,4 ,5
10	ソフトウェアプロセス標準	PRO.imp.3,6, QUA.std.2,3
11	プロセスの改善～概要	PRO.con.7,5,4
12	プロセスの改善～技術	QUA.pro.1,2,P RO.con.6
13	プロセスの改善～標準	QUA.pro.3,4,5, 6,7
14	重要事項のまとめ (各回に時間を分配し、当日の概要及び手法の全体像等について説明することも可)	
15	期末試験	

教科書・参考書	<p>ソフトウェアプロセス, 井上, 飯田, 松本, 共立出版, 2000</p> <p>CMMI標準教本 ~プロセス統合と成果物改善の指針, メアリー ベス 他, JASPIC CMMI V1.1翻訳研究会 訳, 日経BP, 2005</p> <p>共通フレーム2007—経営者、業務部門が参画するシステム開発および取引のために、情報処理推進機構ソフトウェア・エンジニアリングセンター編, オーム社, 2007</p> <p>ソフトウェアでビジネスに勝つ, ワッツ・ハンフリー, 富野 寿 訳, 構造計画研究所, 2003</p> <p>初めて学ぶソフトウェアメトリクス~プロジェクト見積もりのためのデータの導き方, ローレンス・H・パトナム, ウエア・マイヤーズ, 山浦 恒央 訳, 日経BP, 2005</p> <p>Software Metrics: A Rigorous and Practical Approach, Norman Fenton, Shari Lawrence Pfleeger, Course Technology Ptr (Sd); 2nd Ed., 1998.</p> <p>実践的ソフトウェア測定, ジョン・マギヤリー他著, 古山, 富野訳, 構造計画研究所, 2004</p> <p>リファクタリング—プログラムの体質改善テクニック, マーチン ファウラー, 児玉他訳, ピアソンエデュケーション, 2000.</p>
備考	

授業科目名	ヒューマンファクター	
単位数	2	
開設学期	3年後期	
目的	ソフトウェアの使いやすさは、使用する人間に関する知見なくしては開発できない。本講義では、利用者である人間の特性に着目し、使いやすいユーザインタフェースとはどのようなものかを理解しユーザインタフェース設計技法を身に付けることを目標とする	
概要	ヒューマンコンピュータインタフェースは、人間の使う装置やシステムと人間の間の境界に位置し、利用者にとっての使い易さや使い難さ、あるいは誤り難さや誤り易さなどを定める重要な要素である。ヒューマンコンピュータインタフェースの特性は、本来人間の感覚・知覚・認知特性、さらに習熟度等とも密接に関係し、これらの要素に十分配慮して定められるべきものである。このようなヒューマンコンピュータインタフェースに関して、その基本理念、設計指針、最適化、評価法などの問題を考え、学習する。	
目標	<p>人間の感覚や知覚などの特性が理解できる</p> <p>良いインタフェースとはどのようなものが理解できる</p> <p>入出力インタフェースが理解できる</p> <p>ガイドラインとモデルを知り、インタフェース設計に適用できる</p> <p>インタフェースの評価手法が理解できる</p>	
授業展開		学習成果
1	<p>授業の概要と目的、進め方、評価基準について説明する(0.3h)</p> <p>ヒューマンファクターとは何か、ヒューマンファクターとソフトウェアの関係を説明</p>	CMP.cf.7

	する	
2	ヒューマンコンピュータインタフェースとは何か、ヒューマンコンピュータインタフェースの3つの側面とアプローチ方法について説明し、あわせてヒューマンコンピュータインタフェース設計の基本原則について解説する(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.1
3	ヒューマンコンピュータインタフェース設計で重要になるインチ科学の基礎的事項について解説し、メンタルモデルとは何か、メンタルモデルに合致したインタフェース設計について解説する(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.9
4	入力装置と表示装置のインタフェースについて説明し、操作性についても触れる。表示装置の特徴を明らかにし音声によるインタフェースについて説明を加える。最後に視線入力によるインタフェースについても紹介する(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.8
5	ヒューマンコンピュータインタフェースの使いやすさに影響を与える人間とコンピュータとの対話方法に焦点をあて、メッセージをわかりやすく伝えるための手段について解説する(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.2,5,4
6	GUIにおける配慮事項を述べ、GUI設計の留意事項を解説する(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.3
7	ソフトウェアの設計における人間工学的原則と認知科学的原則について述べ、ソフトウェア設計におけるヒューマンコンピュータインタフェースの構造的、機能的視点について解説する(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.7
8	ヒューマンコンピュータインタフェース設計の評価法について解説する	DES.hci.7,VA V.hct.1,2,3,4,5
9	ヒューマンエラーとは何かについて解説し、ヒューマンコンピュータインタフェース設計におけるヒューマンエラーの問題について考え、ヒューマンエラー防止に繋がるインタフェース設計について述べる(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.10
10	インタフェースの国際的な標準を紹介した後、開発者サイドのヒューマンファクターについて解説する(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.6,CM P.cf.8
11	開発者サイドのヒューマンファクターの解説を続ける(冒頭0.1hで前回復習と今回講義の概要)次回講義で行う演習についての課題の事前説明(講義0.2h)	
12	システム提案(1)	
13	システム提案(2)	
14	重要事項のまとめ	
15	期末試験	

教科書・参考書	岡田、西田、葛岡著:「ヒューマンコンピュータインタラクション」ITTextシリーズ、オーム社 村田著:「ヒューマンインタフェースの基礎と応用」日本出版サービス
備考	

授業科目名	開発マネジメント	
単位数	2	
開設学期	3年後期	
目的	ソフトウェア開発プロジェクトの概要を理解し、基本的なビジネススキル、マネジメントスキルを学習することを目的とする	
概要	ソフトウェア開発は個人で実施するのではなく複数のプロフェッショナルが協調して、組織的にプロジェクトとして実施する。ソフトウェア開発プロジェクトを円滑に実施するために、各個人が備えるべき基本的なビジネススキルを具体的に理解し、また、プロジェクトマネジメントスキルの概要を学習することを目的とする。	
目標	ソフトウェア開発者が備えるべき基本的なビジネススキル、及び、基本的なプロジェクトマネジメントスキルを理解し、使えることを目標とする	
授業展開		学習成果
1	授業の概要と目的および進め方について説明する。(0.3h) ソフトウェア開発プロジェクトの概要を理解し、その中でプロジェクトマネジメントの役割を説明する。また、チーム作業と個人の役割を理解する。	MGT.con.1,2,3 ,4
2	チームやグループでの作業でのチームと個人の間で認識すべき問題(複雑性、曖昧性など)とそれへの対処を説明する。	MGT.con.5,6,7 ,8
3	ソフトウェア開発者のコミュニケーションスキル	MGT.con.9,10
4	ソフトウェア開発者のコミュニケーションスキル	MGT.con.11,1 2
5	ソフトウェア開発者のプロフェッショナリズム。個人の行動規範や資格に関して	MGT.con.13,1 4,15
6	ソフトウェア開発者のプロフェッショナリズム。業界の団体やその役割。	PRF.pr.4,5,6,7
7	マネジメントの基礎	MGT.con.20,2 1,22,23,24
8	プロジェクト計画の基礎	MGT.con.25,2 6
9	タイムマネジメント	MGT.con.27,2 8,29
10	リスクマネジメントと品質マネジメント	MGT.con.30,3

		1
11	プロジェクトメンバと組織	MGT.con.32,3 3,34,35,36,37, 38
12	プロジェクトのコントロール	MGT.con.39,4 0,41,42,43,44
13	ソフトウェア構成管理	MGT.con.45,4 6,47,48,49,50, 51
14	重要事項のまとめ	
15	期末試験	
教科書・参 考書	参考書「プロジェクトマネジメント知識体系ガイド 第3版（PMBOK®ガイド第3版）」 発行: PMI	
備考		

付録4 J07-SEの実習科目とインターンシップ、卒業研究

授業科目名	プログラミング基礎実習	
単位数	2	
開設学期	2年後期	
目的	工学的なソフトウェア開発の一環としてプログラムを構築する技術を身につけることを目的とする。	
概要	<p>講師が提示する例題ソフトウェアに対する概略設計内容を入力として、プログラム開発と単体テスト、統合テストを実施する。また、開発したプログラムや実施したテストに関する仕様の文書化も行う。実習にあたり、(主にプログラミングのための)開発支援ツールなどを活用することで、工学的なプログラム開発を体験する。</p> <p>(※注)授業展開や各項目の配分は固定的なものではなく、例題の性質、想定するプログラミングパラダイムなどに応じて柔軟に編成されるべきである。</p>	
目標	ソフトウェアの設計内容が与えられたときに、開発支援ツールなどを活用しながら、主に個人でモジュール分割を行い、それに従ってプログラムを開発し、開発したプログラムをテストする能力を身につける。また、自身のプログラミング作業を管理し、開発したプログラムの仕様書やテスト仕様書を記述する能力を身につける。	
授業展開		学習成果
1	<p>実習の概要と目的および進め方について説明する。</p> <p>実習課題について説明する。</p> <p>実習に用いるツールなどの使用法を理解する。</p> <p>実習で行った作業を振り返る。</p>	
2	<p>設計のレビューを行う。</p> <p>実習で行った作業を振り返る。</p>	
3	<p>モジュール分割を実施する。</p> <p>実習で行った作業を振り返る。</p>	
4	<p>モジュール仕様書をレビューする。</p> <p>単体テストを設計する。</p> <p>実習で行った作業を振り返る。</p>	
5	<p>作業計画を立案する。</p> <p>プログラミングと単体テストを実施する(1)</p> <p>実習で行った作業を振り返る。</p>	
6	<p>プログラミングと単体テストを実施する(2)</p> <p>実習で行った作業を振り返る。</p>	
7	<p>プログラミングと単体テストを実施する(3)</p>	

	実習で行った作業を振り返る.	
8	プログラミングと単体テストを実施する(4)	
	実習で行った作業を振り返る.	
9	モジュールの統合と統合テストを実施する(1)	
	実習で行った作業を振り返る.	
10	モジュールの統合と統合テストを実施する(2)	
	実習で行った作業を振り返る.	
11	プログラムの評価を行う(1)	
	実習で行った作業を振り返る.	
12	プログラムの評価を行う(2)	
	実習で行った作業を振り返る.	
13	成果発表会(1)	
14	成果発表会(2)	
15	全体のまとめ	
教科書・参考書		
備考		

授業科目名	プログラミング応用実習	
単位数	2	
開設学期	3年前期	
目的	工学的なソフトウェア開発の一環としてプログラムを構築する技術を身につけることを目的とする。応用実習では、例題として与えられるドメイン知識の理解に基づいてプログラムを開発する技術や、既存のソフトウェア資産を再利用する技術を習得する。	
概要	講師が提示する例題ソフトウェアシステムに対する要求仕様書を入力として、ソフトウェアアーキテクチャの設計とプログラム開発、テストを実施する。また、実習にあたり、統合開発環境などを活用することで、工学的なプログラム開発を体験する。 (※注)授業展開や各項目の配分は固定的なものではなく、例題のドメインが持つ性質、再利用資産の利用形態などに応じて柔軟に編成されるべきである。	
目標	ソフトウェアの要求仕様が与えられたときに、統合開発環境やプロセス管理ツール、構成管理ツールなどを活用しながら、グループまたは個人でソフトウェアアーキテクチャの設計やプログラム開発、テスト、保守する能力を身につける。また、自身またはグループによる開発作業を管理するとともに開発したソフトウェア成果物の管理をする能力を身につける。	
授業展開		学習成果
1	実習の概要と目的および進め方について説明する。	
	実習課題について説明する。	

	実習に用いるツールなどの使用法を理解する。	
	実習で行った作業を振り返る。	
2	設計のレビューを行う。	
	実習で行った作業を振り返る。	
3	作業計画を立案する。	
	ソフトウェアアーキテクチャを設計する(1)	
	実習で行った作業を振り返る。	
4	ソフトウェアアーキテクチャを設計する(2)	
	実習で行った作業を振り返る。	
5	ソフトウェアアーキテクチャを設計する(3)	
	実習で行った作業を振り返る。	
6	プログラムの設計と構築を行う(1)	
	実習で行った作業を振り返る。	
7	プログラムの設計と構築を行う(2)	
	実習で行った作業を振り返る。	
8	プログラムの設計と構築を行う(3)	
	実習で行った作業を振り返る。	
9	プログラムの設計と構築を行う(4)	
	実習で行った作業を振り返る。	
10	システムテストを実施する(1)	
	実習で行った作業を振り返る。	
11	システムテストを実施する(2)	
	実習で行った作業を振り返る。	
12	ソフトウェアシステムの評価を行う	
	開発実績の評価を行う	
	実習で行った作業を振り返る。	
13	成果発表会(1)	
14	成果発表会(2)	
15	全体のまとめ	
教科書・参考書		
備考		

授業科目名	エンタープライズソフトウェア開発実習
単位数	2 (ただし実習科目のため45時間)

開設学期	3年後期	
目的	これまでに学んだ要求分析、アーキテクチャ設計、ソースコード作成/プログラミング、テストの技法を用いて、永続化データの処理を中心とする実際のエンタープライズソフトウェアシステムを対象として開発工程を通して行い、各工程の作業内容、及び工程間の関係を理解することを目標とする	
概要	講師が提示する例題システムに対する要望を入力として、要求分析、アーキテクチャ設計、ソースコード作成/プログラミング、テストを実施する。グループでの演習を効率良く実施するために実習冒頭でグループのリーダー役を定め、リーダーが中心となって、作業計画と作業目標を実習単位で明確にし、各実習終了前に計画の進捗度合いを確認し、計画に遅延が生じている場合は計画の再調整を実施する。演習にあたりグループの構成員全員が貢献することが望ましい。	
目標	<ul style="list-style-type: none"> ・エンタープライズソフトウェアシステム開発における各工程の作業内容が理解できる ・エンタープライズソフトウェアシステム開発工程のV字の関係が理解できる ・グループでの作業を効率良く進めるためのプロジェクトマネジメントにおける計画マネジメント、コミュニケーションマネジメントを体感する。 	
授業展開		学習成果
1	演習の概要と目的および進め方について説明し、実習のコマ数と各コマの作業目標(想定)を発表する。	グループ演習用のグループ分けも発表する。
	システム要求分析工程およびシステムアーキテクチャ設計工程として、ソフトウェアシステム化の背景と要望を理解し、作業計画と目標を立案する	
2	ソフトウェア要求分析工程の作業成果物を作成する(1)	
3	ソフトウェア要求分析工程の作業成果物を作成する(2)	
4	ソフトウェアアーキテクチャ設計工程における要求からアーキテクチャへの変換タスクの作業成果物を作成する(1)	
5	ソフトウェアアーキテクチャ設計工程における要求からアーキテクチャへの変換タスクの作業成果物を作成する(2)	
6	アーキテクチャ設計工程における最上位設計タスクの作業成果物を作成する(1)	
7	アーキテクチャ設計工程における最上位設計タスクの作業成果物を作成する(2)	
8	データベース設計を実施する(1)	
9	データベース設計を実施する(2)	
10	ソースコード作成及びテスト工程の作業成果物を作成する	

11	テスト工程の作業成果物を作成する(1)	
12	テスト工程の作業成果物を作成する(2)	
13	成果発表会(1)	
14	成果発表会(2)	
15	成果発表会(3)および全体のまとめ	
教科書・参考書	「実践で学ぶソフトウェア開発」 (株)NTTデータソフトウェア工学推進センタ著 オーム社(2008.2)	
備考		

授業科目名	組込みソフトウェア開発実習	
単位数	2 (ただし実習科目のため45時間)	
開設学期	3年後期	
目的	これまでに学んだ要求分析、アーキテクチャ設計、ソースコード作成/プログラミング、テストの技法を用いて実際の組込みソフトウェアシステム開発工程を通して行い、組込み開発特有の制約や性質を加味した各工程の作業内容、及び工程間の関係を理解する。	
概要	講師が提示する例題システムに対する顧客の要望を入力として、組込みソフトウェアシステム開発における要求分析、アーキテクチャ設計、ソースコード作成/プログラミング、テストを実施する。グループでの演習を効率良く実施するために講義冒頭でグループのリーダー役を定め、リーダーが中心となって、作業計画と作業目標を講義単位で明確にし、各講義終了前に計画の進捗度合いを確認し、計画に遅延が生じている場合は計画の再調整を実施する。演習にあたりグループの構成員全員が貢献することが望ましい。	
目標	<ul style="list-style-type: none"> ・組込みソフトウェアシステム開発における各工程の作業内容が理解できる。 ・組込みソフトウェアシステム開発工程のV字の関係が理解できる。 ・グループでの作業を効率良く進めるためのプロジェクトマネジメントにおける計画マネジメント、コミュニケーションマネジメントを体感する。 ・リアルタイム性や省メモリ、クロス開発といった組込み開発特有の性質および制約を理解し、それらを考慮した開発技法を習得する。 	
授業展開		学習成果
1	演習の概要と目的および進め方について説明し、実習のコマ数と各コマの作業目標(想定)を発表する(実習0.3h)	グループ演習用のグループ分けも発表する。
	システム要求分析工程およびシステムアーキテクチャ設計工程として、システム化の背景と要望を理解し、作業計画と目標を立案する	

2	システム要求分析工程およびシステムアーキテクチャ設計工程の作業成果物を作成する	
3	ソフトウェア要求分析工程の作業成果物を作成する(1)	
4	ソフトウェア要求分析工程の作業成果物を作成する(2)	
5	ソフトウェアアーキテクチャ設計工程における要求からアーキテクチャへの変換タスクの作業成果物を作成する(1)	
6	ソフトウェアアーキテクチャ設計工程における要求からアーキテクチャへの変換タスクの作業成果物を作成する(2)	
7	アーキテクチャ設計工程における最上位設計タスクの作業成果物を作成する(1)	
8	アーキテクチャ設計工程における最上位設計タスクの作業成果物を作成する(2)	
9	ソースコード作成及びテスト工程の作業成果物を作成する	
10	テスト工程の作業成果物を作成する(1)	
11	テスト工程の作業成果物を作成する(2)	
12	保守工程の作業成果物を作成する	
13	成果発表会(1)	
14	成果発表会(2)	
15	成果発表会(3)および全体のまとめ	
教科書・参考書	「実践で学ぶソフトウェア開発」 (株)NTTデータソフトウェア工学推進センタ著、オーム社(2008.2) 「組込みソフトウェア開発のためのオブジェクト指向モデリング」 SESSAME WG2著、翔泳社(2006.6) 「組込みソフトウェア開発のための構造化モデリング」 SESSAME WG2著、翔泳社(2006.1) 「組込みソフトウェア開発基礎講座」 杉浦英樹、橋本隆成 著、翔泳社(2007.1)	
備考		

授業科目名	インターンシップ
単位数	2 (ただし演習科目のため、2コマ分、全45時間)
開設学期	3年生夏季休暇
目的	民間企業等との連携により、現実のソフトウェア開発プロジェクトを教材とした実践的な学習を行うことを目的とする。

概要	民間企業等との連携により、現実のソフトウェア開発プロジェクトを教材とした実践的な学習を行う。単なる例題にとどまらない現実のソフトウェア開発作業に関わることにより、納期、予算、組織といった実社会の制約を踏まえたソフトウェア開発の実際について学ぶ。	
目標	開発するシステムの具体化に関わり、ソフトウェアの役割を明確化することができる。企業等の担当者と協調して開発作業に関わり、実開発における諸制約を理解できる。成果について発表、討論ができる。	
授業展開		学習成果
	テーマを選択する際に、インターンシップの目的、担当したい作業とその理由、身につけたい能力とその理由などについて、事前に報告書を作成する。全期間を通じ、各自選択したテーマにおけるソフトウェア開発を連携企業等とともに進める。企業等の担当者、指導教員との議論を通じ、ソフトウェア開発において必要となる思考法、問題解決法について総合的に学ぶ。実社会の制約を踏まえた開発作業を通じて得られた成果および、事前報告書に記載した事項が身についたか、などを事後報告書としてまとめ、発表討論を行う。	
教科書・参考書		
備考		

授業科目名	卒業研究	
単位数	未指定（大学により定めること。学校教育法・大学設置基準。）	
開設学期	4年生前期・後期	
目的	ソフトウェア工学またはソフトウェア工学に関連する特定の課題について研究または制作し、研究の推進方法や知識および独創的な考えの展開能力を習得することを目的とする。	
概要	在学中に修得したソフトウェア工学に関する知識や技術のすべてを、初めての新しい課題に集中して独自の解を求め、その研究成果を論文または制作物の形にまとめ、発表する。これを通して、研究の推進方法、関連研究の調査による知識の進歩、および、新しい独創的な考えを展開してゆく能力を養成する。	
目標	ソフトウェア工学に関する研究の推進方法、関連研究の調査による知識、および、新しい独創的な考えを展開してゆく能力を習得すること。	
授業展開		学習成果
	全期間を通じ、各自選択したテーマにおける研究または制作を指導教員の指導のもとに進める。指導教員等との議論を通じ、特に先進的な研究または制作課題の達成において必要となる創造的思考法、問題解決法について総合的に学ぶ。研究または制作作業を通じて得られた成果を論文または制作物としてまとめ、発表討論を行う。	
教科書・参考書		

備考	
----	--